ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
1 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
2 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
3 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
4 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
5 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
6 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
7 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
8 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
9 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
10 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
11 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
12 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

COPYRIGHT© 1986

ESKAR-2.2

VAX 11/780
MICRODIAG #3

EP-ESKAR-DL-2.2
14 OF 14 JUL 1986
COPYRIGHT© 1986

digital
MADE IN USA

IDENTIFICATION

PRODUCT ID: ESKAR-V2.2

PRODUCT TITLE: ESKAR22 VAX 11/780 MS780-E MICRO DIAGNOSTIC GO CHAIN (VOLUME 3)

DATE: September 1985

MAINTAINED BY: VAX DIAGNOSTIC ENGINEERING

AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

ZZ-ESKAR-V22 Table of Contents
ESKAR3D.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 0002

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1808 MODULE REVISION HISTORY
; 1861 MICRO TRAP HANDLER AND LSI-MONITOR CALL ROUTINES
; 1927 LSI SUPPORT SUBROUTINES
; 1966 RESERVED CONTROL STORE
; 1989 INCREMENT SUBTEST NUMBER
; 1999 SET ARGUMENT COUNT
; 2010 DISPATCH ROUTINE
; 2029 IB WAIT ROUTINE
; 2047 TWINKLE ECO CHECK
; 2084 DISABLE SBI AND FORCE CACHE MISS
; 2097 DISABLE CACHE
; 2108 DISABLE SBI CYCLE
; 2118 TB TEST TRAP ENABLE (DC.DE:)
; 2131 WRITE TB GROUP 1 ROUTINE
; 2152 WRITE TB GROUP 0 ROUTINE
; 2161 WRITE TB 0 AND 1 ROUTINE
; 2170 RANDOM TB WRITE ROUTINE
; 2177 SIMULATE ADDRESS TRANSLATION (RLESUB:)
; 2195 RELOCATE THE VA AND CHECK (RELCHK:)
; 2228 VA MISS IN TB ROUTINE (MISS:)
; 2254 VA HIT IN TB GROUP 0 CHECK (HIT0:)
; 2286 VA CHECK HIT IN GROUP 1 (HIT1:)
; 2295 CONSTAN GENERATORS
; 2353 TB TESTS CONSTANT GENERATOR (TBWPTP:)
; 2436 TARGET SUBROUTINE FOR CALL 3'S (TARGC3:)
; 2454 TEST 171 BRANCH ENABLE A - CURRENT MODE LESS THAN AST LEVEL
; 2564 TEST 172 INSTRUCTION BUFFER WRITE CHECK
; 2820 TEST 173 SAVED CONTEXT AND DATA TYPE QUAD
; 3009 TEST 174 ALU Z BRANCH WITH DATA TYPE QUAD
; 3097 TEST 175 READ.V.NEWPC, WITH TB MISS, TEST
; 3195 TEST 176 READ.V.NEWPC, WITH ACCESS VIOLATION, TEST
; 3292 TEST 177 READ.V.NEWPC, WITH TB PARITY ERROR, TEST
; 3397 TEST 178 RESERVED
; 3402 TEST 179 RESERVED
; 3407 TEST 17A RESERVED
; 3412 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
;1806
;1807 .REGION/1000,13FF
```

```
;1808  .PAGE "MODULE REVISION HISTORY"
;1809 ;*****************************************************************************
;1810 ;
;1811 ;
;1812 ; MODULE NAME: ESKAR3D
;1813 ;
;1814 ; CREATION DATE:  SEPTEMBER 1982
;1815 ; AUTHOR:  DAN GRIGORE
;1816 ;
;1817 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1818 ;
;1819 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1820 ;
;1821 ; -  WHAT CHANGE WAS MADE
;1822 ;
;1823 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1824 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1825 ;
;1826 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1827 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1828 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1829 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1830 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1831 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1832 ;    ESKAR3C.
;1833 ;
;1834 ; START OF REVISION HISTORY:
;1835 ;
;1836 ;  MODIFIED BY: PAUL HAKALA
;1837 ;
;1838 ; 1.0 06-DEC-82: ESKAR40 TEST 18E (Cpu Multi Trans Test)
;1839 ;    added a check for another MS780-E and
;1840 ;    test it if found.
;1841 ;
;1842 ;    ESKAR3D TEST 172  Enable the cache at the
;1843 ;    beginning of test to allow for looping
;1844 ;    on the entire section.
;1845 ;
;1846 ;    ESKAR3E Took out the "?" mark im message
;1847 ;    number 10. This is an illegal Character in
;1848 ;    radix 50 code message "No memory cntrls found"
;1849 ;    now works properly.
;1850 ;
;1851 ;    Did a major cleanup of all documentation in
;1852 ;    all modules.
;1853 ;
;1854 ;
;1855 ;
;1856 ;*****************************************************************************
;1857
```

```
;1858   .PAGE
;1859
```

```
            ;1860   .PAGE
            ;1861   .TOC " MICRO TRAP HANDLER AND LSI-MONITOR CALL ROUTINES"
            ;1862 ;*
            ;1863 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1864 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1865 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1866 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1867 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1868 ;
            ;1869 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS"  MICRO TRAP IT WOULD SET BIT
            ;1870 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1871 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1872 ; R[0F] AND DO A RETURN0.
            ;1873 ;
            ;1874 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1875 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1876 ; TO THE CONSOLE.
            ;1877 ;
            ;1878 ;-
U 1100. 0000.003C.1980.0800.0084.7003   ;1879 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7001   ;1880 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7001   ;1881 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7001   ;1882 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7001   ;1883 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7001   ;1884 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7001   ;1885 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7001   ;1886 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7001   ;1887 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003   ;1888 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003   ;1889 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7001   ;1890 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003   ;1891 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1006   ;1892 TRPH0: Q_ID[USTACK]
U 1006. 0C00.003C.01E0.0800.0000.100B   ;1893  Q_D,D_Q
U 100B. 0000.003C.8180.3C00.0000.100E   ;1894  ID[USTACK]_D
U 100E. 0C00.003C.01E0.0800.0000.1013   ;1895  Q_D,D_Q
U 1013. 0000.003C.01C0.0A78.0000.1016   ;1896  Q_R[0F]
U 1016. 0001.2024.0180.0800.0010.101E   ;1897  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 101E. 0000.013C.0180.0800.0000.1002   ;1898  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000   ;1899 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1900 ;*
            ;1901 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1902 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1903 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1904 ;
            ;1905 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1906 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1907 ;       FAULT STATUS REGISTER
            ;1908 ;       SBI ERROR REGISTER
            ;1909 ;       TIMEOUT ADDRESS REGISTER
            ;1910 ;       MAINTENANCE REGISTER
            ;1911 ;       CACHE PARITY REGISTER
            ;1912 ;       TB ERROR REGISTER 1
            ;1913 ;       TB ERROR REGISTER 0
            ;1914 ;-
```

```
U 1003, 0018,0038,1D80,09E8,0000,1004  ;1915 TRPH1: RC[0D]_SC
U 1004, 0000,003D,D980,0800,0084,7198  ;1916 =0 SETRCF[.9]
U 1005, 0000,003C,49F0,2C00,0000,1022  ;1917  Q_ID[TBER0]
U 1022, 0001,203C,4DF0,2DB0,0000,1026  ;1918  RC[6]_Q,Q_ID[TBER1]
U 1026, 0001,203C,65F0,2DB8,0000,102D  ;1919  RC[7]_Q,Q_ID[SBI.ERR]
U 102D, 0001,203C,6DF0,2DD8,0000,103D  ;1920  RC[0B]_Q,Q_ID[FAULT]
U 103D, 0001,203C,75F0,2DE0,0000,104F  ;1921  RC[0C]_Q,Q_ID[MAINT]
U 104F, 0001,203C,79F0,2DC8,0000,1055  ;1922  RC[9]_Q,Q_ID[PARITY]
U 1055, 0001,203C,69F0,2DC0,0000,105B  ;1923  RC[8]_Q,Q_ID[TIME.ADDR]
U 105B, 0001,203C,81F0,2DD0,0000,1000  ;1924  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001,203D,0180,09F0,0000,1192  ;1925 1000: RC[0E]_Q,ERROR1
```

```
            ;1926  .PAGE
            ;1927  .TOC "    LSI SUPPORT SUBROUTINES"
            ;1928 ;*
            ;1929 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1930 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1931 ;  NEWTST
            ;1932 ;  ERLOOP
            ;1933 ;  ERROR1
            ;1934 ;  ERROR2
            ;1935 ;
U 105F. 0000.003C.65F8.0800.0084.706D  ;1936 SCOPE: SC_K[.10],Q_0   ; SETUP TO WRITE IPL OF 1F
U 106D. 0818.0038.8D80.0800.0000.107D  ;1937  D_K[.1F]    ; ...
U 107D. 0D00.003C.0180.0800.0000.108D  ;1938  D_DAL.SC
U 108D. 0000.003C.3D80.3C00.0000.1109  ;1939  ID[PSL]_D    ; WRITE THE PSL
U 1109. 0818.0038.0580.0800.0000.1154  ;1940  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 1154. 0D00.003C.0180.0800.0000.1184  ;1941  D_DAL.SC
U 1184. 0F00.003C.3180.3C00.0000.1185  ;1942  ID[CES]_D,D_0    ; CLEAR THE CES REGISTER
U 1185. 0000.003C.5D80.3C00.0000.1186  ;1943  ID[ACC.CS]_D    ; SHUT OFF THE FLOATING POINT
U 1186. 0000.003C.3980.3C00.0000.1187  ;1944  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1187. 0000.003C.2980.3C00.0000.1188  ;1945  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1188. 0000.003C.4980.3C00.0000.1189  ;1946  ID[TBER0]_D    ; SHUT OFF THE TB
U 1189. 0818.0038.C180.0800.0000.118A  ;1947  D_K[.FFFF]
U 118A. 0000.003C.6580.3C00.0000.118B  ;1948  ID[SBI.ERR]_D
U 118B. 0F00.003C.6D80.3C00.0000.118C  ;1949  ID[FAULT]_D,D_0
U 118C. 0000.003C.6D80.3C00.0000.118D  ;1950  ID[FAULT]_D
U 118D. 0000.003C.6580.3C00.0000.118E  ;1951  ID[SBI.ERR]_D
U 118E. 0003.003C.0180.09F0.0000.118F  ;1952  RC[0E]_0
U 118F. 0003.003C.0180.0AF8.0000.1190  ;1953  R[0F]_0
U 1190. 0003.003C.0180.09E8.0000.105E  ;1954  RC[0D]_0,J/CALLCON   ; CALL THE CONSOLE
U 1191. 0818.0038.0980.0800.0000.105E  ;1955 ERLOOP: D_K[.2],J/CALLCON   ; ERRLOOP CALL
U 1192. 0810.0038.0180.0978.0000.1193  ;1956 ERROR1: D_RC[0F]
U 1193. 0819.0034.4D80.0800.0000.104E  ;1957  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;1958 104E: D_D.OR.K[.4],J/CALLCON
U 1194. 0810.0038.0180.0978.0000.1195  ;1959 ERROR2: D_RC[0F]
U 1195. 0819.0034.4D80.0800.0000.105A  ;1960  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;1961 105A: D_D.OR.K[.6],J/CALLCON
            ;1962 105E:
            ;1963 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;1964  ID[TXDB]_D,J/CALLCON   ; CALL THE CONSOLE
```

```
            ;1965  .PAGE
            ;1966  .TOC "    RESERVED CONTROL STORE"
            ;1967 ;*
            ;1968 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1969 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1970 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1971 13F1: NOP
U 13F2, 0000.003C.0180.0800.0000.13F3  ;1972 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;1973 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;1974 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;1975 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;1976 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;1977 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;1978 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;1979 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;1980 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;1981 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;1982 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;1983 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;1984 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;1985 13FF: NOP
U 1155, 0000.003C.0180.0800.0000.12AA  ;1986 1155: NOP
U 12AA, 0000.003C.0180.0800.0000.1196  ;1987 12AA: NOP
```

```
            ;1988  .PAGE
            ;1989  .TOC "    INCREMENT SUBTEST NUMBER"
            ;1990 ;*
            ;1991 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1992 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1993 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1994 ; TYPING IT.
            ;1995 ;-
U 1196. 0810,0038,4D80,0978,0000,1197  ;1996 SUBTST: D_RC[0F],KMX/.FF00
U 1197. 0019,4016,4D80,09F8,0000,0001  ;1997  RC[0F]_D+K[.FF00],WORD,RETURN1
```

```
            ;1998  .PAGE
            ;1999  .TOC "    SET ARGUMENT COUNT"
            ;2000 ;*
            ;2001 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2002 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2003 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2004 ;-
U 1198. 0010.0038.01C0.0978.0000.1199  ;2005 SETRCF: Q_RC[0F]
U 1199. 0019.2034.4DC0.0800.0000.119A  ;2006  Q_Q.AND.K[.FF00]
U 119A. 0019.2032.1D80.09F8.0000.0001  ;2007  RC[0F]_Q.OR.SC,RETURN1
            ;2008
```

```
            ;2009  .PAGE
            ;2010  .TOC "   DISPATCH ROUTINE"
U 119B. 0000.003F.0180.0800.0000.1200  ;2011 DISPAT: SUB/SPEC,J/1200
            ;2012
U 1208. 0000.003E.0180.0800.0000.0001  ;2013 1208: RETURN1
U 1251. 0000.003E.0180.0800.0000.0001  ;2014 1251: RETURN1
U 121C. 0000.003E.0180.0800.0000.0001  ;2015 121C: RETURN1
U 124F. 0000.003E.0180.0800.0000.0001  ;2016 124F: RETURN1
U 1253. 0000.003E.0180.0800.0000.0001  ;2017 1253: RETURN1
U 127C. 0018.003A.0DC0.0800.0000.0001  ;2018 127C: Q_K[.3],RETURN1
U 127D. 0018.003A.05C0.0800.0000.0001  ;2019 127D: Q_K[.1],RETURN1
U 127E. 0018.003A.09C0.0800.0000.0001  ;2020 127E: Q_K[.2],RETURN1
U 128F. 0000.003E.01F8.0800.0000.0001  ;2021 128F: Q_0,RETURN1
U 1290. 0000.003E.0180.0800.0000.0001  ;2022 1290: RETURN1
U 12DB. 0000.003E.0180.0800.0000.0001  ;2023 12DB: RETURN1
U 125E. 0000.003E.0180.0800.0000.0001  ;2024 125E: RETURN1
U 1216. 0000.003E.0180.0800.0000.0001  ;2025 1216: RETURN1
U 12DD. 0000.003E.0180.0800.0000.0001  ;2026 12DD: RETURN1
            ;2027
```

```
            ;2028  .PAGE
            ;2029  .TOC "    IB WAIT ROUTINE"
            ;2030 ;*
            ;2031 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2032 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE IB
            ;2033 ; IS COMPLETELY FULL OF DATA.
            ;2034 ;
            ;2035 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2036 ;-
            ;2037
U 119C, 3000,003C,0180,6000,0000,119D  ;2038 IBWAIT: LOAD.IB,IBC/START
U 119D, 0000,003C,0180,F800,0000,119E  ;2039  MCT/ALLOW.IB.READ
U 119E, 0000,003C,0180,F800,0000,119F  ;2040  MCT/ALLOW.IB.READ
U 119F, 0000,003C,0180,F800,0000,11A0  ;2041  MCT/ALLOW.IB.READ
U 11A0, 0000,003C,0180,F800,0000,11A1  ;2042  MCT/ALLOW.IB.READ
U 11A1, 0000,003C,0180,F800,0000,11A2  ;2043. MCT/ALLOW.IB.READ
U 11A2, 1000,003E,0180,F800,0000,0001  ;2044  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2045
```

```
            ;2046  .PAGE
            ;2047  .TOC "    TWINKLE ECO CHECK"
            ;2048 ;*
            ;2049 ;      THE FOLLOWING SUB ROUTINE IS USED TO CHECK IF THE "TWINKLE"  ECO
            ;2050 ; IS INSTALLED ON THIS MACHINE. THIS IS DONE BY CHECKING THE ECO
            ;2051 ; LEVEL OF THE MACHINE AS RECORDED IN THE SYSTEM ID REGISTER. 'TWINKLED'
            ;2052 ; MACHINES WILL HAVE AN ECO LEVEL OF '70' OR GREATER.
            ;2053 ;      THE ROUTINE IS ENTERED WITH THE D REGISTER CONTAINING THE LOOP
            ;2054 ; THAT MUST BE ACHEIVED BEFORE THE ECO LEVEL MUST BE CHECKED. THE LOOP
            ;2055 ; COUNT MUST BE IN RC[0C].
            ;2056 ;
            ;2057 ; THIS ROUTINE MUST BE CALLED WITH A CONSTRAINT OF "=00".
            ;2058 ;
            ;2059 ; THE ROUTINE EXITS WITH THE CONTENTS OF RC[0C] IN THE D REGISTER.
            ;2060 ; A "RETURN1"  IS MADE IF THE LOOP COUNT IS NOT AT THE SPECIFIED LIMIT OR
            ;2061 ; IS IS AND THE TWINKLE ECO IS INSTALLED OTHERWISE, A "RETURN2"
            ;2062 ; IS MADE.
            ;2063 ;-
            ;2064 CHECK_TWINKLE:
U 11A3, 0000,003C,0180,0960,0000,11A4  ;2065  LC_RC[0C]    ; LATCH CURRENT LOOP COUNT
U 11A4, 0011,8000,0180,0800,0010,11A5  ;2066  ALU_D-LC,CLK.UBCC,BYTE  ; AT SPECIFIED LOOP COUNT?
U 11A5, 0000,013C,0180,0800,0000,100C  ;2067  Z?
U 100C, 0000,003C,0180,0800,0000,1007  ;2068 =0 J/CHK_TWINKLE_EX  ; BRANCH IF NO
U 100D, 0000,003C,0DF0,2C00,0000,11A6  ;2069  Q_ID[SYS.ID]    ; GET SYSTEM ECO LEVEL
U 11A6, 0C00,003C,6180,0800,0000,11A7  ;2070  D_Q,K[.F]
U 11A7, 001B,0000,6180,0800,0082,11A8  ;2071  SC_0-K[.F]    ; SETUP TO SHIFT RIGHT
U 11A8, 0D00,003C,0180,0800,0000,1014  ;2072  D_BAL.SC    ; PUT IN BITS<6:0>
U 1014, 0818,0039,8DE0,0800,0000,11FA  ;2073 =0 Q_D,D_K[.1F],CALL[DCF]  ; GENERATE FIELD MASK (= 1FF)
U 1015, 081D,0034,0180,0800,0000,11A9  ;2074  D_D.AND.Q    ; MASK
U 11A9, 0018,0038,F9C0,0800,0000,11AA  ;2075  Q_K[.7E]    ; GENERATE ECO LEVEL OF A "TWINKLED"
U 11AA, 0019,2034,CDC0,0800,0000,11AB  ;2076  Q_Q.AND.K[.F0]    ; MACHINE
U 11AB, 001D,0000,0180,0800,0010,11AC  ;2077  ALU_D-Q,CLK.UBCC  ; IS THIS MACHINE TWINKLED?
U 11AC, 0810,1B38,0180,0960,0000,1007  ;2078  ALU.N?,D_RC[0C]
            ;2079 =0111
            ;2080 CHK_TWINKLE_EX:
U 1007, 0000,003E,0180,0800,0000,0001  ;2081  RETURN1     ; BRANCH IF YES
U 100F, 0000,003E,0180,0800,0000,0002  ;2082  RETURN2     ; NO
```

```
            ;2083  .PAGE
            ;2084  .TOC "    DISABLE SBI AND FORCE CACHE MISS"
            ;2085 ;**
            ;2086 ; THIS SUBROUTINE IS CALLED TO SET THE DISABLE SBI CYCLE
            ;2087 ; AND FORCE MISS BITS IN THE CACHE MAINTENANCE REGISTER.
            ;2088 ; UPON RETURNING THE PREVIOUS CONTENTS OF D, Q AND SC
            ;2089 ; ARE LOST.
            ;2090 ;--
U 11AD, 0F18.0038.C1F8.0800.0000.1018  ;2091 CSOFF: D_0,Q_0,ALU_K[.FFFF]
U 1018, 0500.003D.0380.0800.0000.11F4  ;2092 =0 D_D.LEFT,SI/7,CALL,J/DC9
U 1019, 0000.003C.8580.0800.0084.71AE  ;2093  SC_K[.C]
U 11AE, 0D00.003C.0180.0800.0000.11AF  ;2094  D_DAL.SC
U 11AF, 0000.003E.7580.3C00.0000.0001  ;2095  ID[MAINT]_D.RETURN1
```

```
            ;2096  .PAGE
            ;2097  .TOC "    DISABLE CACHE"
            ;2098 ;**
            ;2099 ; THIS SUBROUTINE IS CALLED TO SET THE FORCE CACHE MISS
            ;2100 ; BITS FOR EACH GROUP IN THE MAINTENANCE REGISTER ON THE ID BUS.
            ;2101 ;--
U 11B0. 0818,0038,0DF8,0800,0000,11B1  ;2102 COFF: D_K[.3],Q_0
U 11B1. 0000,003C,6180,0800,0084,71B2  ;2103  SC_K[.F]
U 11B2. 0D00,003C,0180,0800,0000,11B3  ;2104  D_DAL.SC
U 11B3. 0000,003E,7580,3C00,0000,0001  ;2105  ID[MAINT]_D,RETURN1
            ;2106
```

```
            :2107  .PAGE
            :2108  .TOC "    DISABLE SBI CYCLE"
            :2109 ;**
            :2110 ; THIS SUBROUTINE IS CALLED TO DISABLE SBI CYCLES BY SETTING THE
            :2111 ; APPROPRIATE BIT IN THE SBI MAINTENANCE REGISTER.
            :2112 ;--
U 11B4. 0000.003C.8580.0800.0084.71B5  ;2113 SOFF: SC_K[.C]
U 11B5. 0803.001C.0180.0800.0000.11B6  ;2114  ALU_NOT.MASK,D_ALU
U 11B6. 0000.003E.7580.3C00.0000.0001  ;2115  ID[MAINT]_D.RETURN1
            :2116
```

```
            ;2117  .PAGE
            ;2118  .TOC "    TB TEST TRAP ENABLE (DC.DE:)"
            ;2119 ;+*
            ;2120 ; THIS ROUTINE GENERATES A CONSTANT WHICH IS
            ;2121 ; USED IN THE TB TESTS TO SET THE FLAG WHICH ENABLES
            ;2122 ; CERTAIN TRAPS.
            ;2123 ;--
            ;2124 =0
U 101A. 0000,003D,0180,0800,0000,11F8  ;2125 DC.DE: CALL,J/DCD
U 101B. 0000,003C,0180,0800,0000,101C  ;2126  NOP
U 101C. 0000,003D,0180,0800,0000,11F9  ;2127 =0 CALL,J/DCE
U 101D. 0000,003E,0180,0800,0000,0001  ;2128  RETURN1
            ;2129
```

```
            :2130  .PAGE
            :2131  .TOC "    WRITE TB GROUP 1 ROUTINE"
            :2132 ;**
            :2133 ; THESE SUBROUTINES ARE USED TO WRITE THE TB. R[8]
            :2134 ; IS ASSUMED TO CONTAIN THE PTE TO BE WRITTEN. THE
            :2135 ; VA SHOULD BE SET UP BEFORE CALLING THESE ROUTINES.
            :2136 ; MME WILL BE SET HERE. UPON RETURN THE CONTENTS
            :2137 ; OF TBER1 IS LEFT IN Q. THE PREVIOUS CONTENTS OF
            :2138 ; D AND Q ARE LOST.
            :2139 ; THIS ROUTINE WRITES GROUP 1
            :2140 ;--
            :2141 =0
U 1020. 0F00.003D.0180.0800.0000.11F3  ;2142 TBWR1: D_0,CALL,J/DC8
U 1021. 0100.003C.0180.0800.0000.1024  ;2143 TBWR: D_D.LEFT2,SI/ZERO
U 1024. 0100.003D.0180.0800.0000.11EB  ;2144 =0 D_D.LEFT2,SI/ZERO,CALL,J/DC0
U 1025. 0100.003C.0180.0800.0000.102E  ;2145  D_D.LEFT2,SI/ZERO
U 102E. 0100.003D.0180.0800.0000.11EC  ;2146 =0 D_D.LEFT2,SI/ZERO,CALL,J/DC1
U 102F. 0800.003C.4980.3E40.0000.11B7  ;2147 TBWRA: ID[TBER0]_D,D_R[8]
U 11B7. 0000.003C.4180.3C00.0000.11B8  ;2148  ID[TBUF]_D
U 11B8. 0000.003E.4DF0.2C00.0000.0001  ;2149  Q_ID[TBER1],RETURN1
            :2150
```

```
            ;2151   .PAGE
            ;2152   .TOC "     WRITE TB GROUP 0 ROUTINE"
            ;2153 ;++
            ;2154 ; ROUTINE TO WRITE GROUP 0
            ;2155 ;--
            ;2156 =0
U 1030, 0F00.003D.0180.0800.0000.11EF  ;2157 TBWR0: D_0.CALL.J/DC4
U 1031, 0000.003C.0180.0800.0000.1021  ;2158  J/TBWR
            ;2159
```

```
            :2160  .PAGE
            :2161  .TOC "    WRITE TB 0 AND 1 ROUTINE"
            :2162 ;**
            :2163 ; ROUTINE TO WRITE BOTH GROUP 0 AND 1
            :2164 ;--
            :2165 =0
U 1032. 0F00.003D.0180.0800.0000.11F3  :2166 TBWRB: D_0,CALL,J/DC8
U 1033. 0500.003C.0180.0800.0000.1021  :2167  D_D.LEFT,SI/ZERO,J/TBWR
            :2168
```

```
            ;2169  .PAGE
            ;2170  .TOC "    RANDOM TB WRITE ROUTINE"
            ;2171 ;**
            ;2172 ; THIS ROUTINE WRITES THE TB RANDOMLY.
            ;2173 ;--
U 11B9, 0818,0038,0580,0800,0000,102F  ;2174 TBWRAN: D_K[.1],J/TBWRA
            ;2175
```

```
            ;2176  .PAGE
            ;2177  .TOC "    SIMULATE ADDRESS TRANSLATION (RLESUB:)"
            ;2178 ;**
            ;2179 ; THIS ROUTINE IS USED TO SIMULATE AN ADDRESS TRANSLATION.
            ;2180 ; THE VIRTUAL ADDRESS SHOULD BE PLACED IN R[5], THE PTE IN
            ;2181 ; R[6] AND THE RESULTING PHYSICAL ADDRESS (AS IT WOULD APPEAR IN THE
            ;2182 ; TIME OUT ADDRESS REGISTER) WILL BE LEFT IN
            ;2183 ; R[7]. THE PREVIOUS CONTENTS OF D, Q AND SC ARE LOST.
            ;2184 ;--
            ;2185 =0
U 103E. 0800.003D.0180.0A30.0000.11EB  ;2186 RELSUB: D_R[6],CALL,J/DC0
U 103F. 0500.003C.D980.0A28.0084.7040  ;2187  D_D.LEFT,SI/ZERO,SC_K[.9],LAB_R[5]
U 1040. 0003.0011.01C0.0800.0000.11EB  ;2188 =0 ALU_MASK+1,Q_ALU,CALL,J/DC0
U 1041. 001C.0024.01C0.0800.0000.11BA  ;2189  ALU_LA[ANDNOT]Q,Q_ALU
U 11BA. 081D.0030.0180.0800.0000.11BB  ;2190  ALU_D[OR]Q,D_ALU
U 11BB. 0200.003C.0180.0800.0000.11BC  ;2191  D_D.RIGHT2,SI/ZERO
U 11BC. 0001.003E.0180.0ABB.0000.0001  ;2192  R[7]_D,RETURN1
            ;2193
```

```
            ;2194  .PAGE
            ;2195  .TOC "    RELOCATE THE VA AND CHECK (RELCHK:)"
            ;2196 ;**
            ;2197 ; THIS SUBROUTINE IS CALLED TO RELOCATE THE VA, GET THE
            ;2198 ; TIME OUT ADDRESS REGISTER AND COMPARE IT TO THE CONTENTS OF R(7).
            ;2199 ; RETURN3 IF THEY ARE EQUAL; RETURN 2 IF THEY ARE NOT EQUAL;
            ;2200 ; AND RETURN1 IF A UTRAP OCCURS. THE R(F) TRAP FLAGS WILL BE SET AS
            ;2201 ; INDICATED IN R(E).
            ;2202 ;--
U 11BD. 0000.003C.0180.0A70.0000.11BE  ;2203 RELCHK: LAB_R(0E)
U 11BE. 0000.003C.0180.52F8.0000.11BF  ;2204  R(0F)_LA,FS/MCT,MCT/READ.V.WCHK ; DO THE REFERNCE
U 11BF. 0800.003C.69F0.2E78.0000.11C0  ;2205  Q_ID(TIME.ADDR),D_R(0F)
U 11C0. 0018.0038.1980.0AF8.0000.11C1  ;2206  ALU_0(K),R(0F)_ALU
U 11C1. 0000.003C.0180.0A70.0000.11C2  ;2207  LAB_R(0E)
U 11C2. 001C.2000.0180.0800.0010.11C3  ;2208  ALU_LA(A-B)D,CLK.UBCC  ; SEE IF A TRAP OCCURRED.
U 11C3. 0000.013C.0180.0800.0000.1042  ;2209  Z?
U 1042. 0001.003C.0180.09C8.0000.11CC  ;2210 =0 RC(9)_D,J/TBTRUE ; TRAPPED SO RETURN1.
U 1043. 0000.003C.0180.0800.0000.1048  ;2211  NOP
U 1048. 0F00.003D.0180.0800.0000.11F9  ;2212 =0 D_0,CALL,J/DCE   ; MASK OFF THE UPPER 4 BITS
U 1049. 0500.003C.0180.0800.0000.11C4  ;2213  D_D.LEFT,SI/ZERO
U 11C4. 0001.003C.0180.0800.0082.11C5  ;2214  SC_D
U 11C5. 0803.0010.0180.0800.0000.11C6  ;2215  ALU_MASK+1,D_ALU
U 11C6. 001D.2024.01C0.0800.0000.11C7  ;2216  ALU_Q(ANDNOT)D,Q_ALU
U 11C7. 0800.003C.0180.0A38.0000.11C8  ;2217  D_R(7)    ; SEE IF ADDRESS RELOCATED
U 11C8. 0001.003C.0180.09E0.0000.11C9  ;2218  RC(0C)_D
U 11C9. 0001.203C.0180.09D8.0000.11CA  ;2219  RC(0B)_Q
U 11CA. 001D.2000.0180.0800.0010.11CB  ;2220  ALU_Q-D,CLK.UBCC  ; PROPERLY.
U 11CB. 0000.013C.0180.0800.0000.104A  ;2221  Z?
U 104A. 0000.003E.0180.0800.0000.0002  ;2222 =0 RETURN2    ; INCORRECT.
U 104B. 0000.003E.0180.0800.0000.0003  ;2223  RETURN3    ; OK
U 11CC. 0000.003C.4DF0.2C00.0000.11CD  ;2224 TBTRUE: Q_ID(TBER1)
U 11CD. 0001.203E.0180.09F0.0000.0001  ;2225  RC(0E)_Q,RETURN1
            ;2226
```

```
            ;2227  .PAGE
            ;2228  .TOC "    VA MISS IN TB ROUTINE (MISS:)"
            ;2229 ;**
            ;2230 ;
            ;2231 ; THIS SUBROUTINE IS CALLED TO SEE IF THE VA IS A MISS IN THE TB.
            ;2232 ; IF NO TRAP OCCURS WHEN THE REFERNCE IS MADE EXECUTE A RETURN1.
            ;2233 ; IF A NON TB MISS TRAP OCCURS ON THE REFERENCE THEN EXECUTE
            ;2234 ; A RETURN2. IF A TB MISS TRAP OCCURS RETURN3.
            ;2235 ;
            ;2236 ;--
U 11CE. 0000.003C.0180.0A70.0000.11CF  ;2237 MISS: LAB_R[0E]
U 11CF. 0000.003C.0180.52F8.0000.11D0  ;2238  R[0F]_LA,FS/MCT,MCT/READ.V.WCHK ; SHOULD TRAP HERE IF MISS
U 11D0. 0800.003C.0180.0A78.0000.11D1  ;2239  D_R[0F]
U 11D1. 0018.0038.1980.0AF8.0000.11D2  ;2240  ALU_0(K),R[0F]_ALU
U 11D2. 0000.003C.0180.0A70.0000.11D3  ;2241  LAB_R[0E]
U 11D3. 0001.003C.0180.09C8.0000.11D4  ;2242  RC[9]_D
U 11D4. 001C.2000.0180.0800.0010.11D5  ;2243  ALU_LA[A-B]D,CLK.UBCC  ; SEE IF A TRAP OCCRRED.
U 11D5. 0000.013C.0180.0800.0000.104C  ;2244  Z?
U 104C. 0000.003C.0180.0800.0000.11D6  ;2245 =0 J/MISSA
U 104D. 0000.003E.0180.0800.0000.0001  ;2246  RETURN1    ; NO TRAP SO RETURN1.
U 11D6. 0000.003C.D580.0800.0084.71D7  ;2247 MISSA: SC_K[.6]    ; IF A TRAP OCCURRED SEE
U 11D7. 0001.0024.0180.0800.0010.11D8  ;2248  ALU_D.ANDNOT.MASK,CLK.UBCC ; IF IT WAS A TB MISS TRAP.
U 11D8. 0000.013C.4DF0.2C00.0000.1056  ;2249  Z?,Q_ID[TBER1]
U 1056. 0001.203E.0180.09F0.0000.0002  ;2250 =0 RETURN2,RC[0E]_Q  ; NOT TB MISS.
U 1057. 0000.003E.0180.0800.0000.0003  ;2251  RETURN3    ; TB MISS.
            ;2252
```

```
            ;2253  .PAGE
            ;2254  .TOC "    VA HIT IN TB GROUP 0 CHECK (HIT0:)"
            ;2255 ;++
            ;2256 ;
            ;2257 ; THESE SUBROUTINES ARE CALLED TO SEE IF THE VA IS A HIT.
            ;2258 ; RETURN1 IF A NON TB MISS TRAP OCCURS. RETURN2 IF A TB MISS
            ;2259 ; TRAP OCCURS. RETURN3 IF NO TRAP OCCURS BUT THE APPROPRIATE
            ;2260 ; HIT/MISS BIT IN TBER0 IS INCORRECT.
            ;2261 ; RETURN4 IF NO TRAP OCCURS AND THE HIT/MISS BIT IS SET.
            ;2262 ; CHECK HIT IN GROUP 0.
            ;2263 ;
            ;2264 ;--
U 11D9. 0000.003C.D580.0A70.0084.71DA  ;2265 HIT0: LAB_R[0E],SC_K[.6]
U 11DA. 0000.003C.0180.52F8.0000.11DB  ;2266 HIT: R[0F]_LA,FS/MCT,MCT/READ.V.WCHK  ; NO TRAP IF HIT
U 11DB. 0800.003C.49F0.2E78.0000.11DC  ;2267  Q_ID[TBER0],D_R[0F]
U 11DC. 0018.0038.1980.0AF8.0000.11DD  ;2268  ALU_0(K),R[0F]_ALU
U 11DD. 0001.203C.0180.09E8.0000.11DE  ;2269  RC[0D]_Q
U 11DE. 0003.001C.0180.09F0.0000.11DF  ;2270  RC[0E]_ALU,ALU_NOT.MASK
U 11DF. 0000.003C.0180.0A70.0000.11E0  ;2271  LAB_R[0E]
U 11E0. 001C.2000.0180.0800.0010.11E1  ;2272  ALU_LA[A-B]D,CLK.UBCC   ; SEE IF TRAP OCCURRED.
U 11E1. 0001.013C.0180.09C8.0000.1058  ;2273  Z?,RC[9]_D
U 1058. 0000.003C.0180.0800.0000.11E3  ;2274 =0 J/HITA    ; TRAP OCCURRED.
U 1059. 0001.2024.0180.0800.0010.11E2  ;2275  ALU_Q.ANDNOT.MASK,CLK.UBCC ; NO TRAP BUT IS HIT/MISS BIT SET?
U 11E2. 0000.013C.0180.0800.0000.105C  ;2276  Z?
U 105C. 0000.003E.0180.0800.0000.0004  ;2277 =0 RETURN4    ; SET SO IT WAS A HIT.
U 105D. 0000.003E.0180.0800.0000.0003  ;2278  RETURN3    ; CLEAR SO REFERENCE WAS A MISS.
U 11E3. 0000.003C.D580.0800.0084.71E4  ;2279 HITA: SC_K[.6]    ; TRAPPED, SO SEE IF TB MISS TRAP.
U 11E4. 0001.0024.0180.0800.0010.11E5  ;2280  ALU_D.ANDNOT.MASK,CLK.UBCC
U 11E5. 0000.013C.0180.0800.0000.1060  ;2281  Z?
U 1060. 0000.003C.0180.0800.0000.11CC  ;2282 =0 J/TBTRUE    ; NOT TB MISS TRAP.
U 1061. 0000.003E.0180.0800.0000.0002  ;2283  RETURN2    ; TB MISS TRAP OCCURRED.
            ;2284
```

```
            ;2285  .PAGE
            ;2286  .TOC "    VA CHECK HIT IN GROUP 1 (HIT1:)"
            ;2287 ;**
            ;2288 ;
            ;2289 ; CHECK HIT IN GROUP 1.
            ;2290 ;
            ;2291 ;--
U 11E6. 0000,003C,5D80,0A70,0084,71DA  ;2292 HIT1: LAB_R[0E],SC_K[.7],J/HIT
            ;2293
```

```
            ;2294  .PAGE
            ;2295  .TOC "    CONSTAN GENERATORS"
            ;2296 ;**
            ;2297 ;
            ;2298 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER.
            ;2299 ;
            ;2300 ;--
U 11E7, 0118.0038.1B80.0800.0000.11FB  ;2301 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 11E8, 0118.0038.0B80.0800.0000.11FB  ;2302 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 11E9, 0118.0038.0780.0800.0000.11FB  ;2303 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 11EA, 0118.0038.C380.0800.0000.11FB  ;2304 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
U 11EB, 0018.0038.1980.0800.0000.11E7  ;2305 DC0: ALU_0(K),J/S00
U 11EC, 0018.0038.1980.0800.0000.11E8  ;2306 DC1: ALU_0(K),J/S01
U 11ED, 0018.0038.1980.0800.0000.11E9  ;2307 DC2: ALU_0(K),J/S10
U 11EE, 0018.0038.1980.0800.0000.11EA  ;2308 DC3: ALU_0(K),J/S11
U 11EF, 0018.0038.0980.0800.0000.11E7  ;2309 DC4: ALU_K[.2],J/S00
U 11F0, 0018.0038.0980.0800.0000.11E8  ;2310 DC5: ALU_K[.2],J/S01
U 11F1, 0018.0038.0980.0800.0000.11E9  ;2311 DC6: ALU_K[.2],J/S10
U 11F2, 0018.0038.0980.0800.0000.11EA  ;2312 DC7: ALU_K[.2],J/S11
U 11F3, 0018.0038.0580.0800.0000.11E7  ;2313 DC8: ALU_K[.1],J/S00
U 11F4, 0018.0038.0580.0800.0000.11E8  ;2314 DC9: ALU_K[.1],J/S01
U 11F5, 0018.0038.0580.0800.0000.11E9  ;2315 DCA: ALU_K[.1],J/S10
U 11F6, 0018.0038.0580.0800.0000.11EA  ;2316 DCB: ALU_K[.1],J/S11
U 11F7, 0018.0038.C180.0800.0000.11E7  ;2317 DCC: ALU_K[.FFFF],J/S00
U 11F8, 0018.0038.C180.0800.0000.11E8  ;2318 DCD: ALU_K[.FFFF],J/S01
U 11F9, 0018.0038.C180.0800.0000.11E9  ;2319 DCE: ALU_K[.FFFF],J/S10
U 11FA, 0018.0038.C180.0800.0000.11EA  ;2320 DCF: ALU_K[.FFFF],J/S11
U 11FB, 0100.003E.0380.0800.0000.0001  ;2321 SX: D_D.LEFT2,SI/7,RETURN1
U 1200, 0018.0038.01C0.0800.0000.1062  ;2322 DC.5: Q_K[.8]
            ;2323 =0
U 1062, 0000.003D.0180.0800.0000.11F0  ;2324 DC.5A: CALL,J/DC5
U 1063, 0019.2000.05C0.0800.0010.1201  ;2325  ALU_Q-K[.1],Q_ALU,CLK.UBCC
U 1201, 0000.013C.0180.0800.0000.106E  ;2326  Z?
U 106E, 0000.003C.0180.0800.0000.1062  ;2327 =0 J/DC.5A
U 106F, 0000.003E.0180.0800.0000.0001  ;2328  RETURN1
            ;2329 =0
U 1070, 0000.003D.0180.0800.0000.1200  ;2330 DC.A: CALL,J/DC.5
U 1071, 0500.003E.0180.0800.0000.0001  ;2331  D_D.LEFT,SI/ZERO,RETURN1
U 1202, 0018.0038.01C0.0800.0000.1072  ;2332 DC.9: Q_K[.8]
            ;2333 =0
U 1072, 0000.003D.0180.0800.0000.11F4  ;2334 DC.9A: CALL,J/DC9
U 1073, 0019.2000.05C0.0800.0010.1203  ;2335  ALU_Q-K[.1],Q_ALU,CLK.UBCC
U 1203, 0000.013C.0180.0800.0000.107E  ;2336  Z?
U 107E, 0000.003C.0180.0800.0000.1072  ;2337 =0 J/DC.9A
U 107F, 0000.003E.0180.0800.0000.0001  ;2338  RETURN1
            ;2339 =0
U 1080, 0000.003D.0180.0800.0000.1202  ;2340 DC.C: CALL,J/DC.9
U 1081, 0018.0038.C180.0800.0000.1204  ;2341  ALU_K[.FFFF]
U 1204, 0600.003E.0380.0800.0000.0001  ;2342  D_D.RIGHT,SI/7,RETURN1
            ;2343 =0
U 1082, 0000.003D.0180.0800.0000.1202  ;2344 DC.6: CALL,J/DC.9
U 1083, 0018.0038.0580.0800.0000.1205  ;2345  ALU_K[.1]
U 1205, 0100.003E.0380.0800.0000.0001  ;2346  D_D.LEFT2,SI/7,RETURN1
            ;2347 =0
U 108E, 0000.003D.0180.0800.0000.1202  ;2348 DC.3: CALL,J/DC.9
```

```
U 108F, 0018,0038,C180,0800,0000,1206  ;2349  ALU_K[.FFFF]
U 1206, 0500,003E,0380,0800,0000,0001  ;2350  D_D.LEFT,SI/7,RETURN1
          ;2351
```

```
                ;2352  .PAGE
                ;2353  .TOC "    TB TESTS CONSTANT GENERATOR (TBWPTP:)"
                ;2354 ;**
                ;2355 ;
                ;2356 ; THIS SUBROUTINE IS USED THROUGHOUT THE TB TESTS TO GENERATE
                ;2357 ; A SET OF CONSTANTS. UPON RETURNING FROM THIS ROUTINE THE
                ;2358 ; SCRATCH PAD REGISTERS HAVE BEEN SET UP TO CONTAIN THE
                ;2359 ; FOLLOWING DATA PATTERNS:
                ;2360 ;
                ;2361 ;  R[0B]  7FFFC000
                ;2362 ;  R[0C]  A4000000
                ;2363 ;  R[0D]  001FFFFF
                ;2364 ;
                ;2365 ;  RC[0]  A4155555
                ;2366 ; RC[1]  4CCCC000
                ;2367 ; RC[2]  0AAAAA80
                ;2368 ;
                ;2369 ;  RC[3]  A4199999
                ;2370 ;  RC[4]  55554000
                ;2371 ;  RC[5]  0CCCCC80
                ;2372 ;
                ;2373 ;  RC[6]  A40AAAAA
                ;2374 ;  RC[7]  33330000
                ;2375 ;  RC[8]  05555500
                ;2376 ;
                ;2377 ;  R[0]  A4066666
                ;2378 ;  R[1]  2AAA8000
                ;2379 ;  R[2]  03333300
                ;2380 ;
                ;2381 ;--
U 1207. 0F18.0038.0580.0800.0000.1209  ;2382 TBWPTP: D_0,ALU_K[.1]
U 1209. 0218.0038.0B80.0800.0000.120A  ;2383  D_D.RIGHT2,SI/7,ALU_K[.2]
U 120A. 0218.0038.0B80.0800.0000.120B  ;2384  D_D.RIGHT2,SI/7,ALU_K[.2]
U 120B. 0200.003C.0380.0800.0000.120C  ;2385  D_D.RIGHT2,SI/7
U 120C. 0001.003C.0180.0AE0.0000.120D  ;2386  R[0C]_D
U 120D. 0000.003C.6180.0800.0084.720E  ;2387  SC_K[.F]
U 120E. 0043.0010.0180.0AD8.0000.120F  ;2388  ALU_MASK+1,R[0B]_ALU.RIGHT.SI/ZERO
U 120F. 081B.0000.0580.0800.0000.1210  ;2389  D_0-K[.1]
U 1210. 0018.0038.21C0.0800.0000.1211  ;2390  Q_K[.14]
U 1211. 0019.2014.0580.0800.0082.1212  ;2391  ALU_Q+K[.1],SC_ALU
U 1212. 0003.001C.01C0.0800.0000.1213  ;2392  ALU_NOT.MASK,Q_ALU
U 1213. 001D.2014.0180.0AE8.0000.1090  ;2393  ALU_Q[A+B]D,R[0D]_ALU
U 1090. 0000.003D.0180.0800.0000.1200  ;2394 =0 CALL,J/DC.5
U 1091. 0000.003C.0180.0A68.0000.1214  ;2395  LAB_R[0D]
U 1214. 081C.2034.0180.0800.0000.1215  ;2396  ALU_LA[AND]D,D_ALU
U 1215. 081C.2030.0180.0A60.0000.1217  ;2397  LAB_R[0C],ALU_LA[OR]D,D_ALU
U 1217. 0001.003C.0180.0980.0000.1092  ;2398  RC[0]_D
U 1092. 0001.003D.0180.0AB0.0000.1080  ;2399 =0 R[6]_D,CALL,J/DC.C
U 1093. 0000.003C.0180.0A58.0000.1218  ;2400  LAB_R[0B]
U 1218. 081C.2034.0180.0988.0000.1098  ;2401  ALU_LA[AND]D,D_ALU,RC[1]_ALU
U 1098. 0001.003D.0180.0AA8.0000.103E  ;2402 =0 R[5]_D,CALL,J/RELSUB
U 1099. 0001.003C.0180.0990.0000.109A  ;2403  RC[2]_D
U 109A. 0000.003D.0180.0800.0000.1202  ;2404 =0 CALL,J/DC.9
U 109B. 0000.003C.0180.0A68.0000.1219  ;2405  LAB_R[0D]
U 1219. 081C.2034.0180.0800.0000.121A  ;2406  ALU_LA[AND]D,D_ALU
```

```
U 121A. 081C.2030.0180.0A60.0000.121B  ;2407  LAB_R[0C],ALU_LA[OR]D,D_ALU
U 121B. 0001.003C.0180.0998.0000.109C  ;2408  RC[3]_D
U 109C. 0001.003D.0180.0AB0.0000.1200  ;2409 =0 R[6]_D,CALL,J/DC.5
U 109D. 0000.003C.0180.0A58.0000.121D  ;2410  LAB_R[0B]
U 121D. 081C.2034.0180.09A0.0000.109E  ;2411  ALU_LA[AND]D,D_ALU,RC[4]_ALU
U 109E. 0001.003D.0180.0AA8.0000.103E  ;2412 =0 R[5]_D,CALL,J/RELSUB
U 109F. 0001.003C.0180.09A8.0000.10A0  ;2413  RC[5]_D
U 10A0. 0000.003D.0180.0800.0000.1070  ;2414 =0 CALL,J/DC.A
U 10A1. 0000.003C.0180.0A68.0000.121E  ;2415  LAB_R[0D]
U 121E. 081C.2034.0180.0800.0000.121F  ;2416  ALU_LA[AND]D,D_ALU
U 121F. 081C.2030.0180.0A60.0000.1220  ;2417  LAB_R[0C],ALU_LA[OR]D,D_ALU
U 1220. 0001.003C.0180.09B0.0000.10A2  ;2418  RC[6]_D
U 10A2. 0001.003D.0180.0AB0.0000.108E  ;2419 =0 R[6]_D,CALL,J/DC.3
U 10A3. 0000.003C.0180.0A58.0000.1221  ;2420  LAB_R[0B]
U 1221. 081C.2034.0180.09B8.0000.10A4  ;2421  ALU_LA[AND]D,D_ALU,RC[7]_ALU
U 10A4. 0001.003D.0180.0AA8.0000.103E  ;2422 =0 R[5]_D,CALL,J/RELSUB
U 10A5. 0001.003C.0180.09C0.0000.10A6  ;2423  RC[8]_D
U 10A6. 0000.003D.0180.0800.0000.1082  ;2424 =0 CALL,J/DC.6
U 10A7. 0000.003C.0180.0A68.0000.1222  ;2425  LAB_R[0D]
U 1222. 081C.2034.0180.0800.0000.1223  ;2426  ALU_LA[AND]D,D_ALU
U 1223. 081C.2030.0180.0A60.0000.1224  ;2427  LAB_R[0C],ALU_LA[OR]D,D_ALU
U 1224. 0001.003C.0180.0A80.0000.10A8  ;2428  R[0]_D
U 10A8. 0001.003D.0180.0AB0.0000.1070  ;2429 =0 R[6]_D,CALL,J/DC.A
U 10A9. 0000.003C.0180.0A58.0000.1225  ;2430  LAB_R[0B]
U 1225. 081C.2034.0180.0A88.0000.10AA  ;2431  ALU_LA[AND]D,D_ALU,R[1]_ALU
U 10AA. 0001.003D.0180.0AA8.0000.103E  ;2432 =0 R[5]_D,CALL,J/RELSUB
U 10AB. 0001.003E.0180.0A90.0000.0001  ;2433  R[2]_D,RETURN1
          ;2434
```

```
            ;2435  .PAGE
            ;2436  .TOC "    TARGET SUBROUTINE FOR CALL 3'S (TARGC3:)"
            ;2437 ;**
            ;2438 ;
            ;2439 ; THIS IS A TARGET SUBROUTINE FOR CALL 3'S WHICH ARE EXECUTED
            ;2440 ; WITHOUT IRD TO TEST IBUF INITIATED FUNCTIONS IN THE TB
            ;2441 ; WHICH COULD RESULT IN ERROR CONDITIONS.
            ;2442 ;
            ;2443 ;--
U 1226. 0000.003F.0180.0800.0000.117C  ;2444 TARGC3: SUB/SPEC
U 117C. 0000.003E.0180.0800.0000.0001  ;2445 117C: RETURN1
U 117D. 0000.003E.0180.0800.0000.0002  ;2446 117D: RETURN2
U 117E. 0000.003E.0180.0800.0000.0003  ;2447 117E: RETURN3
U 117F. 0000.003E.0180.0800.0000.0004  ;2448 117F: RETURN4
U 11FC. 0000.003D.0180.0800.0000.1192  ;2449 11FC: ERROR1
U 11FD. 0000.003D.0180.0800.0000.1192  ;2450 11FD: ERROR1
U 11FE. 0000.003D.0180.0800.0000.1192  ;2451 11FE: ERROR1
U 11FF. 0000.003D.0180.0800.0000.1192  ;2452 11FF: ERROR1
            ;2453
```

```
;2454 .PAGE "TEST 171 BRANCH ENABLE A - CURRENT MODE LESS THAN AST LEVEL"
;2455 ;*************************************************************************
;2456 ;**
;2457 ; TEST 171 BRANCH ENABLE A -- CURRENT MODE LESS THAN AST LEVEL
;2458 ;
;2459 ; TEST DESCRIPTION
;2460 ; THIS TEST CHECKS BRANCH ENABLE A BY RUNNING A COUNT PATTERN
;2461 ; THRU THE CURRENT MODE BITS FOR ALL 8 LEVELS OF AST. IF A FAILURE
;2462 ; IS DETECTED, IT MEANS THAT THE BRANCH WAS INCORRECT FOR THE CURRENT
;2463 ; VALUES OF THE AST LEVEL AND CURRENT MODE. FOLLOWING IS A TABLE OF
;2464 ; THE EXPECTED BRANCH CONDITION FOR EACH VALUE TESTED:
;2465 ;
;2466 ; AST LEVEL CURRENT MODE  BRANCH CONDITION
;2467 ; --------- ------------  ----------------
;2468 ;
;2469 ;     00      00 - 11   FALSE
;2470 ;     01    00    TRUE
;2471 ;      01 - 11    FALSE
;2472 ;     10      00 - 01   TRUE
;2473 ;      10 - 11    FALSE
;2474 ;     11      00 - 10   TRUE
;2475 ;    11    FALSE
;2476 ;    100      00 - 11   TRUE
;2477 ;    101      00 - 11   TRUE
;2478 ;    110      00 - 11   TRUE
;2479 ;    111      00 - 11   TRUE
;2480 ;
;2481 ; LOGIC DESCRIPTION
;2482 ; THIS TEST CHECKS THE CURRENT MODE/AST LEVEL COMPARATOR AND THE
;2483 ; BRANCH BIT MULTIPLEXOR ON THE ICL MODULE, AND THE BRANCH BIT
;2484 ; MULTIPLEXOR ON THE USC MODULE.
;2485 ;
;2486 ; ERROR DESCRIPTION
;2487 ; DATA: CURRENT AST LEVEL
;2488 ;       CURRENT CURRENT MODE
;2489 ;
;2490 ; SYNC POINT DESCRIPTION
;2491 ; ADDRESS SYNC1E--BRANCH IS ENABLED
;2492 ;--
;2493 ;*************************************************************************
;2494 =0
U 10AC, 0000.003D.0180.0800.0000.105F  ;2495 T171: NEWTST
U 10AD, 0018.0038.0980.09F8.0000.1227  ;2496  RC[0F]_K[.2]
U 1227, 0003.003C.0180.09F0.0000.1228  ;2497  RC[0E]_0    ; INIT AST LEVEL
U 1228, 0003.003C.0180.09E8.0000.1229  ;2498  RC[0D]_0    ; INIT CURRENT MODE (FOR TYPEOUT)
U 1229, 0003.003C.0180.0A80.0000.122A  ;2499  R[0]_0     ; INIT THE BRANCH TRUE FLAG
U 122A, 0818.0038.0580.0800.0000.122B  ;2500  D_K[.1]     ; GENERATE AN INCREMENT VALUE
U 122B, 0B00.003C.0180.0800.0000.122C  ;2501  D_D.SWAP    ; FOR THE PSL CURRENT MODE
U 122C, 0001.003C.0180.0A88.0000.122D  ;2502  R[1]_D     ; SAVE IN R1
U 122D, 0003.003C.0180.0A90.0000.122E  ;2503  R[2]_0     ; INIT VALUE OF CURRENT MODE FOR PSL
;2504
;2505 ;////////////////////////////////////////////////////////////////////////
;2506 ;*
;2507 ; START THE TEST
;2508 ;-
```

```
            ;2509
U 122E. 0018.0038.0180.09D8.0000.122F  ;2510  RC[0B]_K[.8]    ; SET THE LOOP COUNT FOR AST LEVEL
U 122F. 0018.0038.1180.09E0.0000.1230  ;2511  RC[0C]_K[.4]    ; SET THE LOOP COUNT FOR CURRENT MODE
U 1230. 0810.0038.0180.0970.0000.1231  ;2512 T171L1: D_RC[0E]    ; GET CURRENT AST LEVEL
U 1231. 0000.003C.3180.3C00.0000.1232  ;2513  ID[CES]_D    ; LOAD THE AST LEVEL
U 1232. 0800.003C.0180.0A10.0000.1233  ;2514  D_R[2]    ; GET THE CURRENT MODE
U 1233. 0000.003C.3D80.3C00.0000.10AE  ;2515  ID[PSL]_D    ; LOAD THE PSL
U 10AE. 0000.003D.0180.0800.0000.1191  ;2516 =0 ERLOOP
U 10AF. 0000.003C.0180.0800.0000.10B0  ;2517  NOP
U 10B0. 0000.003D.0180.0800.0000.1247  ;2518 =0 CALL,J/TSTRE1    ; GO EXECUTE THE BRANCH
U 10B1. 001C.8000.0180.0800.0010.1234  ;2519  ALU_LA[A-B]Q,CLK.UBCC,BYTE ; CHECK THE BRANCH RESULT
U 1234. 0000.013C.0180.0800.0000.10B2  ;2520  Z?
U 10B2. 0000.003D.0180.0800.0000.1194  ;2521 =0 ERROR2    ; BRANCH FAILED
            ;2522
            ;2523 ;*
            ;2524 ; THE FOLLOWING CODE IS LOOP CONTROL FOR THIS TEST
            ;2525 ;-
            ;2526
U 10B3. 0810.0038.0180.0968.0000.1235  ;2527  D_RC[0D]    ; INCREMENT THE CURRENT MODE
U 1235. 0019.0014.0580.09E8.0000.1236  ;2528  RC[0D]_D+K[.1]    ; FOR TYPEOUT
U 1236. 0800.003C.0180.0A10.0000.1237  ;2529  D_R[2]    ; INCREMENT THE CURRENT MODE FOR
U 1237. 0000.003C.0180.0A08.0000.1238  ;2530  LAB_R[1]    ; THE PSL
U 1238. 001C.2014.0180.0A90.0000.1239  ;2531  R[2]_LA+D    ; ...
U 1239. 0810.0038.0180.0960.0000.123A  ;2532  D_RC[0C]
U 123A. 0019.8000.0580.09E0.0010.123B  ;2533  RC[0C]_D-K[.1],CLK.UBCC,BYTE ; CHECK CURRENT MODE LOOP COUNT
U 123B. 0000.013C.0180.0800.0000.10B4  ;2534  Z?
U 10B4. 0000.003C.0180.0800.0000.1243  ;2535 =0 J/T171P1    ; GO DETERMINE BRANCH RESULT
U 10B5. 0003.003C.0180.0A90.0000.123C  ;2536  R[2]_0    ; INIT THE CURRENT MODE
U 123C. 0003.003C.0180.09E8.0000.123D  ;2537  RC[0D]_0    ; INIT CURRENT MODE FOR TYPEOUT
U 123D. 0018.0038.1180.09E0.0000.123E  ;2538  RC[0C]_K[.4]    ; INIT CURRENT MODE LOOP COUNT
U 123E. 0810.0038.0180.0970.0000.123F  ;2539  D_RC[0E]    ; GET AST LEVEL
U 123F. 0019.0014.0580.09F0.0000.1240  ;2540  RC[0E]_D+K[.1]    ; INCREMENT IT
U 1240. 0810.0038.0180.0958.0000.1241  ;2541  D_RC[0B]
U 1241. 0019.8000.0580.09D8.0010.1242  ;2542  RC[0B]_D-K[.1],CLK.UBCC,BYTE ; CHECK AST LEVEL LOOP COUNT
U 1242. 0000.013C.0180.0800.0000.10B6  ;2543  Z?
U 10B6. 0000.003C.0180.0800.0000.1243  ;2544 =0 J/T171P1    ; CONTINUE
U 10B7. 0000.003C.0180.0800.0000.1248  ;2545  J/T171END    ; END TEST
U 1243. 0810.0038.0180.0970.0000.1244  ;2546 T171P1: D_RC[0E]    ; GET AST LEVEL
U 1244. 0010.0038.01C0.0968.0000.1245  ;2547  Q_RC[0D]    ; GET CURRENT MODE
U 1245. 001D.A000.0180.0800.0010.1246  ;2548  ALU_Q-D,CLK.UBCC,BYTE  ; SEE IF CM LSS AST LEVEL
U 1246. 0000.1B3C.0180.0800.0000.1017  ;2549  ALU.N?
U 1017. 0003.003C.0180.0A80.0000.1230  ;2550 =0111 R[0]_0,J/T171L1    ; CM GTE AST
U 101F. 0018.0038.0580.0A80.0000.1230  ;2551  R[0]_K[.1],J/T171L1  ; CM LSS AST
            ;2552
            ;2553 ;*
            ;2554 ; THIS ROUTINE EXECUTES THE BRANCH ENABLE A. IT RETURNS WITH A ZERO OR
            ;2555 ; ONE IN THE Q REGISTER DEPENDING ON WHETHER THE BRANCH BIT ASSERTED OR NOT.
            ;2556 ;-
            ;2557
            ;2558 SYNC1E:
U 1247. 0000.0A3C.0180.0800.0000.1023  ;2559 TSTRE1: MODE.LSS.ASTLVL?  ; EXECUTE THE BRANCH
U 1023. 0000.003E.01F8.0A00.0000.0001  ;2560 =011 Q_0,LAB_R[0],RETURN1  ; BRANCH BIT DID NOT ASCERT
U 1027. 0018.003A.05C0.0A00.0000.0001  ;2561  Q_K[.1],LAB_R[0],RETURN1 ; BRANCH BIT ASCERTED
U 1248. 0000.003C.0180.0800.0000.10B8  ;2562 T171END:NOP
            ;2563
```

```
                ;2564 .PAGE "TEST 172 INSTRUCTION BUFFER WRITE CHECK"
                ;2565 ;*****************************************************************
                ;2566 ;**
                ;2567 ;
                ;2568 ; TEST 172 INSTRUCTION BUFFER WRITE CHECK
                ;2569 ;
                ;2570 ; TEST DESCRIPTION
                ;2571 ;
                ;2572 ; THIS TEST CHECKS THAT THE "IB WRITE CHECK"  SIGNAL IS GENERATED
                ;2573 ; CORRECTLY AND THAT IT PERFORMS THE CORRECT FUNCTION ON THE
                ;2574 ; TRANSLATION BUFFER. BIT 9 OF TB ERROR REGISTER 0 IS ALSO TESTED.
                ;2575 ;
                ;2576 ; SUBTEST 1 - CHECK "IB WRITE CHECK FALSE"  IN VAX MODE
                ;2577 ;
                ;2578 ; SUBTEST 2 - CHECK "IB WRITE CHECK TRUE"  IN VAX MODE
                ;2579 ;
                ;2580 ; SUBTEST 3 - CHECK "IB WRITE CHECK FALSE"  IN COMPATABILITY MODE
                ;2581 ;
                ;2582 ; SUBTEST 4 - CHECK "IB WRITE CHECK TRUE"  IN COMPATABILITY MODE
                ;2583 ;
                ;2584 ; LOGIC DESCRIPTION
                ;2585 ;
                ;2586 ; THIS TEST CHECKS THE "IB WRITE CHECK"  FLIP FLOP ON THE IRC MODULE
                ;2587 ; AND THE CIRCUIT THAT RECEIVES THE SIGNAL ON THE TBM MODULE. BIT 9
                ;2588 ; OF TB ERROR REGISTER 0, ON THE TBM MODULE, IS ALSO TESTED.
                ;2589 ;
                ;2590 ; ERROR DESCRIPTION
                ;2591 ;
                ;2592 ; ERR1X: EXPECTED TB ERROR REG 0
                ;2593 ;        RECEIVED TB ERROR REG 0
                ;2594 ;  LOOP COUNT FOR SUBTESTS 3 AND 4
                ;2595 ; ERR2X: EXPECTED MICRO TRAP FLAG
                ;2596 ;        RECEIVED MICRO TRAP FLAG
                ;2597 ;--
                ;2598 ;*****************************************************************
                ;2599
                ;2600 =0
U 10B8, 0000.003D.0180.0800.0000.105F  ;2601 T004: NEWTST
U 10B9, 0000.003C.75F0.2C00.0000.1249  ;2602  Q_ID[MAINT]    ; Get current maintenance reg
U 1249, 0000.003C.6580.0800.0084.724A  ;2603  SC_K[.10]    ; Set up to clear bit 16
U 124A, 0801.2034.0180.0800.0000.124B  ;2604  D_Q.AND.MASK    ; Clear bit 16
U 124B, 0000.003C.7580.3C00.0000.124C  ;2605  ID[MAINT]_D    ; Reload the register enable cache
U 124C, 0018.0038.0980.09F8.0000.124D  ;2606  RC[0F]_K[.2]
U 124D, 0000.003C.2180.0800.0084.724E  ;2607  SC_K[.14]    ; GENERATE DATA FOR TB ER0
U 124E, 0803.001C.0180.0800.0000.1250  ;2608  D_NOT.MASK
U 1250, 0819.0030.0580.0800.0000.1252  ;2609  D_D.OR.K[.1]
U 1252, 0000.003C.4980.3C00.0000.1254  ;2610  ID[TBER0]_D    ; LOAD THE REGISTER
U 1254, 0818.0038.3180.0800.0000.1255  ;2611  D_K[.40]
U 1255, 0019.0014.1180.0A80.0200.1256  ;2612  R[0]_D+K[.4],VA_ALU    ; LOAD THE VA
U 1256, 0818.0038.CD80.0800.0000.1257  ;2613  D_K[.F0]    ; GENERATE A PTE FOR READ ONLY ACCESS
U 1257, 0819.0014.0180.0800.0000.1258  ;2614  D_D+K[.8]    ; ...
U 1258, 0B00.003C.0180.0800.0000.1259  ;2615  D_D.SWAP
U 1259, 0000.003C.4180.3C00.0000.125A  ;2616  ID[TBUF]_D    ; LOAD THE PTE
U 125A, 0818.0038.4180.0800.0000.125B  ;2617  D_K[.80]    ; GENERATE MASK FOR TBER0
U 125B, 0100.003C.0180.0800.0000.125C  ;2618  D_D.LEFT2
```

```
U 125C. 0001.003C.0180.0A88.0000.10BA  ;2619  R[1]_D    ; SAVE = 200
          ;2620
          ;2621 ;////////////////////////////////////////////////////////////////////////
          ;2622 ;*
          ;2623 ; FIRST CHECK THAT IB WRITE CHECK DOES NOT ASCERT FOR A "HALT"  INSTRUCTION
          ;2624 ;-
          ;2625
U 10BA. 0000.003D.0180.0800.0000.1196  ;2626 =0 SUBTEST
U 10BB. 0003.003C.0180.09F0.0000.10BC  ;2627  RC[0E]_0    ; SET EXPECTED TBER0 DATA
U 10BC. 0000.003D.0180.0800.0000.1191  ;2628 =0 ERLOOP
U 10BD. 0000.003C.0180.0800.0000.10BE  ;2629  NOP
U 10BE. 2000.003D.0180.0A00.4200.119C  ;2630 =0 ALU_R[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 10BF. 0000.003C.0180.0800.0000.10C0  ;2631  NOP
U 10C0. 0000.003D.0180.0800.0000.119B  ;2632 =0 CALL,J/DISPAT    ; LATCH THE IB WRITE CHECK BIT
U 10C1. 0000.003C.0180.F800.0000.125D  ;2633  MCT/ALLOW.IB.READ   ; LATCH THE ERROR REGISTER
U 125D. 0000.003C.49F0.2C00.0000.125F  ;2634  Q_ID[TBER0]    ; READ THE ERROR REGISTER
U 125F. 001C.0034.01C0.0A08.0010.1260  ;2635  Q_Q.AND.R[1],CLK.UBCC   ; CHECK BIT 9
U 1260. 0003.013C.0180.0800.0200.10C2  ;2636  Z?,VA_ALU,ALU_0(A)   ; BRANCH IF OK
          ;2637 =0
U 10C2. 0001.203D.0180.09E8.0000.1194  ;2638 ERR10: ERROR2,RC[0D]_Q    ; TBER0 BIT 9 IS NOT CLEAR
          ;2639
          ;2640 ;*
          ;2641 ; NOW CHECK THAT A "READ.V.IBCHK"  FUNCTION DOES NOT TRAP
          ;2642 ;-
          ;2643
U 10C3. 0000.003C.0180.0A00.0200.1261  ;2644  ALU_R[0],VA_ALU    ; RELOAD THE VA
U 1261. 0000.003C.0980.0800.0084.7262  ;2645  SC_K[.2]    ; GENERATE EXPECTED DATA
U 1262. 0003.001C.0180.09F0.0000.1263  ;2646  RC[0E]_NOT.MASK
U 1263. 0003.001C.0180.0AF8.0000.1264  ;2647  R[0F]_NOT.MASK    ; SET TRAP FLAG
U 1264. 0000.003C.0180.5800.0000.1265  ;2648  MCT/READ.V.IBCHK   ; EXECUTE THE FUNCTION
U 1265. 000D.003C.01C0.0A78.0000.1266  ;2649  Q_R[0F]     ; GET THE TRAP FLAG
U 1266. 0810.0038.0180.0970.0000.1267  ;2650  D_RC[0E]    ; GET THE EXPECTED DATA
U 1267. 001D.2000.0180.0800.0010.1268  ;2651  ALU_Q-D,CLK.UBCC   ; CHECK FOR NO TRAP
U 1268. 000D.013C.0180.0800.0000.10C4  ;2652  Z?
          ;2653 =0
U 10C4. 0001.203D.0180.09E8.0000.1194  ;2654 ERR20: ERROR2,RC[0D]_Q    ; MICRO TRAP OCCURRED
U 10C5. 0000.003C.0180.0800.0000.10C6  ;2655  NOP
          ;2656
          ;2657 ;////////////////////////////////////////////////////////////////////////
          ;2658 ;*
          ;2659 ; CHECK THAT A "CLRB"  INSTRUCTION CAUSES IB WRITE CHECK TO ASCERT
          ;2660 ;-
          ;2661
U 10C6. 0000.003D.0180.0800.000D.1196  ;2662 =0 SUBTEST
U 10C7. 0800.003C.0180.0A08.0000.1269  ;2663  D_R[1]     ; GET THE EXPECTED TBER0 DATA
U 1269. 0001.003C.0180.09F0.000D.126A  ;2664  RC[0E]_D    ; SAVE
U 126A. 0800.003C.0180.0A00.000D.126B  ;2665  D_R[0]     ; GET IB ADDRESS
U 126B. 0019.0014.1180.0A80.000D.10C8  ;2666  R[0]_D+K[.4]    ; INCREMENT
U 10C8. 0000.003D.0180.0800.0000.1191  ;2667 =0 ERLOOP
U 10C9. 0000.003C.0180.0800.000D.10CA  ;2668  NOP
U 10CA. 2000.003D.0180.0A00.4200.119C  ;2669 =0 ALU_R[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 10CB. 0810.0038.0180.0970.0000.10CC  ;2670  D_RC[0E]    ; GET EXPECTED DATA
U 10CC. 0000.003D.0180.0800.0000.119B  ;2671 =0 CALL,J/DISPAT    ; LATCH THE WRITE CHECK SIGNAL
U 10CD. 0000.003C.0180.F800.0000.126C  ;2672  MCT/ALLOW.IB.READ   ; LATCH THE ERROR REGISTER
U 126C. 0000.003C.49F0.2C00.0000.126D  ;2673  Q_ID[TBER0]    ; READ THE ERROR REG
```

```
U 126D. 001C.0034.01C0.0A08.0000.126E  ;2674  Q_Q.AND.R[1]   ; MASK
U 126E. 001D.2000.0180.0800.0010.126F  ;2675  ALU_Q-D,CLK.UBCC  ; CHECK
U 126F. 0000.013C.0180.0800.0000.10CE  ;2676  Z?
        ;2677 =0
U 10CE. 0001.203D.0180.09E8.0000.1194  ;2678 ERR11: ERROR2,RC[0D]_Q   ; IB WRITE CHECK DID NOT SET IN ERR REG
        ;2679
        ;2680 ;*
        ;2681 ; NOW CHECK THAT A PROTECTION MICRO TRAP OCCURS WITH A READ.V.IBCHK FUNCTION
        ;2682 ;-
        ;2683
U 10CF. 0000.003C.0180.0A00.0200.1270  ;2684  VA_ALU,ALU_R[0]   ; RELOAD THE ADDRESS
U 1270. 0000.003C.1180.0800.0084.7271  ;2685  SC_K[.4]   ; GENERATE THE EXPECTED DATA
U 1271. 0003.003C.0180.09F0.0000.1272  ;2686  RC[0E]_0
U 1272. 0003.001C.0180.0AF8.0000.1273  ;2687  R[0F]_NOT.MASK   ; SET THE TRAP FLAG
U 1273. 0000.003C.0180.5800.0000.1274  ;2688  MCT/READ.V.IBCHK  ; EXECUTE THE FUNCTION
U 1274. 0000.003C.01C0.0A78.0010.1275  ;2689  Q_R[0F],CLK.UBCC  ; GET AND CHECK THE TRAP FLAG
U 1275. 0000.013C.0180.0800.0000.10D0  ;2690  Z?
        ;2691 =0
U 10D0. 0001.203D.0180.09E8.0000.1194  ;2692 ERR21: ERROR2,RC[0D]_Q   ; MICRO TRAP DID NOT OCCUR
U 10D1. 0000.003C.0180.0800.0000.10D2  ;2693  NOP
        ;2694
        ;2695
        ;2696 ;////////////////////////////////////////////////////////////////////////
        ;2697 ;*
        ;2698 ; CHECK THAT IB WRITE CHECK DOES NOT SET IN COMPATABILITY MODE
        ;2699 ;-
        ;2700
U 10D2. 0000.003D.0180.0800.0000.1196  ;2701 =0 SUBTEST
U 10D3. 0000.003C.0180.0800.0000.10D4  ;2702  NOP
U 10D4. 0000.003D.0D80.0800.0084.7198  ;2703 =0 SETRCF[.3]
U 10D5. 0018.0038.1180.09E0.0000.1276  ;2704  RC[0C]_K[.4]   ; SET THE LOOP COUNT
U 1276. 0800.003C.0180.0A00.0000.1277  ;2705  D_R[0]   ; INCREMENT IB ADDRESS TO DATA
U 1277. 0019.0014.1180.0A80.0000.1278  ;2706  R[0]_D+K[.4]   ; FOR THIS SUBTEST
U 1278. 0003.003C.0180.09F0.0000.10D6  ;2707  RC[0E]_0   ; SET THE EXPECTED DATA
U 10D6. 0000.003D.0180.0800.0000.1191  ;2708 =0 ERLOOP
U 10D7. 0000.003C.8D80.0800.0084.7279  ;2709 NEWL1: SC_K[.1F]   ; GENERATE DATA FOR PSL
U 1279. 0000.003C.3DF0.2C00.0000.127A  ;2710  Q_ID[PSL]   ; GET THE PSL
U 127A. 0801.201C.0180.0800.0000.127B  ;2711  D_Q.ORNOT.MASK   ; INSER THE CM BIT
U 127B. 0000.003C.3D80.3C00.0000.10D8  ;2712  ID[PSL]_D   ; SET COMPATABILITY MODE
U 10D8. 2000.003D.0180.0A00.4200.119C  ;2713 =0 ALU_R[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 10D9. 0000.003C.0180.0800.0000.10DA  ;2714  NOP
U 10DA. 0000.003D.0180.0800.0000.119B  ;2715 =0 CALL,J/DISPAT   ; LATCH THE IB WRITE CHECK SIGNAL
U 10DB. 0000.003C.0180.F800.0000.127F  ;2716  MCT/ALLOW.IB.READ  ; LATCH THE ERROR REGISTER
U 127F. 0000.003C.49F0.2C00.0000.1280  ;2717  Q_ID[TBER0]   ; READ THE ERROR REGISTER
U 1280. 001C.0034.01C0.0A08.0010.1281  ;2718  Q_Q.AND.R[1],CLK.UBCC  ; MASK AND CHECK
U 1281. 0000.013C.0180.0800.0000.10DC  ;2719  Z?
        ;2720 =0
U 10DC. 0001.203D.0180.09E8.0000.1194  ;2721 ERR12: ERROR2,RC[0D]_Q   ; TBER0 BIT 9 SHOULD BE CLEAR
U 10DD. 0800.003C.0180.0A00.0000.1282  ;2722  D_R[0]   ; GET THE IB ADDRESS
U 1282. 0019.0014.1180.0A80.0000.1283  ;2723  R[0]_D+K[.4]   ; SET THE NEXT OP CODE
U 1283. 0810.0038.0180.0960.0000.1284  ;2724  D_RC[0C]   ; GET THE LOOP COUNT
        ;2725  RC[0C]_D-K[.1],D_ALU,
U 1284. 0819.0000.0580.09E0.0010.1285  ;2726       CLK.UBCC   ; DONE WITH LOOP?
U 1285. 0000.013C.0180.0800.0000.10DE  ;2727  Z?
U 10DE. 0000.003C.0180.0800.0000.1008  ;2728 =0 J/NEWL1_CHK_ECO   ; CONTINUE LOOPING
```

```
U 10DF, 0000,003C,0180,0800,0000,10E0  ;2729  J/T004_S4
        ;2730 =00
        ;2731 NEWL1_CHK_ECO:
U 1008, 0818,0039,0580,0800,0000,11A3  ;2732  D_K[.1],CALL[CHECK_TWINKLE] ; CHECK FOR TWINKLE ECO
U 1009, 0000,003C,0180,0800,0000,10D7  ;2733  J/NEWL1    ; BRANCH IF LOOP COUNT NEQ 1
        ;2734       ; OR IT IS AND TWINKLE INSTALLED
U 100A, 0000,003C,0180,0800,0000,10E0  ;2735  NOP    ; DON'T EXECUTE LOOP COUNT EQL 1
        ;2736 =
        ;2737
        ;2738 ;////////////////////////////////////////////////////////////////////////
        ;2739 ;*
        ;2740 ; CHECK THAT IB WRITE CHECK SETS IN COMPATABILITY MODE
        ;2741 ;-
        ;2742 =0
        ;2743 T004_S4:
U 10E0, 0000,003D,0180,0800,0000,1196  ;2744  SUBTEST
U 10E1, 0800,003C,0180,0A08,0000,1286  ;2745  D_R[1]
U 1286, 0001,003C,0180,09F0,0000,1287  ;2746  RC[0E]_D    ; SET THE EXPECTED DATA
U 1287, 0018,003B,D580,09E0,0000,10E2  ;2747  RC[0C]_K[.6]    ; SET THE LOOP COUNT
U 10E2, 0000,003D,0180,0800,0000,1191  ;2748 =0 ERLOOP
U 10E3, 0000,003C,8D80,0800,0084,7288  ;2749 NEWL2: SC_K[.1F]    ; GENERATE DATA FOR PSL
U 1288, 0000,003C,3DF0,2C00,0000,1289  ;2750  Q_ID[PSL]    ; GET THE PSL
U 1289, 0801,201C,0180,0800,0000,128A  ;2751  D_Q.ORNOT.MASK    ; INSER THE CM BIT
U 128A, 0000,003C,3D80,3C00,0000,10E4  ;2752  ID[PSL]_D    ; SET COMPATABILITY MODE
U 10E4, 2000,003D,0180,0A00,4200,119C  ;2753 =0 ALU_R[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 10E5, 0000,003C,0180,0800,0000,10E6  ;2754  NOP
U 10E6, 0000,003D,0180,0800,0000,119B  ;2755 =0 CALL,J/DISPAT    ; LATCH THE IB WRITE CHECK BIT
U 10E7, 0000,003C,0180,F800,0000,128B  ;2756  MCT/ALLOW.IB.READ    ; LATCH THE ERROR REGISTER
U 128B, 0000,003C,49F0,2C00,0000,128C  ;2757  Q_ID[TBER0]    ; READ THE ERROR REGISTER
U 128C, 001C,0034,01C0,0A08,0000,128D  ;2758  Q_Q.AND.R[1]    ; MASK
U 128D, 0810,003B,0180,0970,0000,128E  ;2759  D_RC[0E]    ; GET THE EXPECTED DATA
U 128E, 001D,2000,0180,0800,0010,1291  ;2760  ALU_Q-D,CLK.UBCC    ; CHECK
U 1291, 0000,013C,0180,0800,0000,10E8  ;2761  Z?
        ;2762 =0
U 10E8, 0001,203D,0180,09E8,0000,1194  ;2763 ERR13: ERROR2,RC[0D]_Q    ; IB WRITE CHECK DID NOT SET
U 10E9, 0800,003C,0180,0A00,0000,1292  ;2764  D_R[0]    ; GET IB ADDRESS
U 1292, 0019,0014,1180,0A80,0000,1293  ;2765  R[0]_D+K[.4]    ; SELECT NEXT PATTERN
U 1293, 0810,003B,0180,0960,0000,1294  ;2766  D_RC[0C]    ; GET THE LOOP COUNT
U 1294, 0019,0000,0580,09E0,0010,1295  ;2767  RC[0C]_D-K[.1],CLK.UBCC    ; DONE LOOPING?
U 1295, 0000,013C,0180,0800,0000,10EA  ;2768  Z?
U 10EA, 0000,003C,0180,0800,0000,1010  ;2769 =0 J/NEWL2_CHK_ECO    ; CHECK FOR TWINKLE ECO
U 10EB, 0000,003C,0180,0800,0000,1012  ;2770  J/T004_END_TEST    ; EXIT TEST
        ;2771 =00
        ;2772 NEWL2_CHK_ECO:
U 1010, 0818,0039,1180,0800,0000,11A3  ;2773  D_K[.4],CALL[CHECK_TWINKLE] ; CHECK FOR ECO PRESENT
U 1011, 0000,003C,0180,0800,0000,10E3  ;2774  J/NEWL2    ; EITHER NOT AT LOOP COUNT EQL 4
        ;2775       ; OR TWINKLE ECO IS PRESENT
        ;2776 T004_END_TEST:
U 1012, 0000,003C,0180,0800,0000,10EC  ;2777  NOP
        ;2778 =
        ;2779
        ;2780 ; FOLLOWING IS THE IB DATA FOR THE ABOVE TEST:
        ;2781 ;% 44
        ;2782 ; SUBTEST 1
        ;2783 ;$ 0000,0000
```

```
;2784 ;
;2785 ; SUBTEST 2
;2786 ;$ 0094,0000
;2787 ;
;2788 ; SUBTEST 3
;2789 ;$ 0000,0000
;2790 ;$ 3000,0000
;2791 ; OCTAL OF ABOVE HEX DATA = 30000 (BIT INSTRUCTION)
;2792 ;
;2793 ;$ 0BC0,0000
;2794 ; OCTAL OF ABOVE HEX DATA = 5700 (TST INSTRUCTION)
;2795 ;
;2796 ; TWINKLE ECO
;2797 ;*************************************************
;2798 ;$ 0D40,0000
;2799 ; OCTAL OF ABOVE HEX DATA = 6500 (MFPI INSTRUCTION)
;2800 ;*************************************************
;2801
;2802 ; SUBTEST 4
;2803 ;$ 0A00,0000
;2804 ; OCTAL = 005000
;2805 ;
;2806 ;$ 69C0,0000
;2807 ; OCTAL OF ABOVE HEX DATA = 64700 (ADD INSTRUCTION)
;2808 ; TWINKLE ECO
;2809 ;*************************************************
;2810 ;$ 0DC0,0000
;2811 ; OCTAL = 6700 (SXT)
;2812 ;$ 0C40,0000
;2813 ; OCTAL = 6100 (ROL)
;2814 ;$ 1D40,0000
;2815 ; OCTAL = 16500 (MOV)
;2816 ;$ 0940,0000
;2817 ; OCTAL = 4700 (JSR)
;2818 ;*************************************************
;2819
```

```
                ;2820 .PAGE "TEST 173 SAVED CONTEXT AND DATA TYPE QUAD"
                ;2821 ;*****************************************************************************
                ;2822 ;**
                ;2823 ;
                ;2824 ; TEST 173 SAVED CONTEXT AND DATA TYPE QUAD
                ;2825 ;
                ;2826 ; TEST DESCRIPTION
                ;2827 ;
                ;2828 ; THIS TEST CHECKS THE UNALIGNED AND PAGE TRAPS FOR DATA TYPE QUAD
                ;2829 ; USING THE CURRENT CONTEXT AND THE SAVED CONTEXT.
                ;2830 ;
                ;2831 ; SUBTEST 1 - CURRENT CONTEXT = QUAD. CHECK THE UNALIGNED TRAP FOR
                ;2832 ;      VA VALUES OF 0, 1, AND 2.
                ;2833 ;
                ;2834 ; SUBTEST 2 - SAVED CONTEXT = QUAD, CURRENT CONTEXT = WORD. CHECK THE
                ;2835 ;      UNALIGNED TRAP FOR A VA VALUE OF 1.
                ;2836 ;
                ;2837 ; SUBTEST 3 - CURRENT CONTEXT = QUAD. CHECK THE PAGE TRAP WITH VA
                ;2838 ;      VALUES OF 1F8, 1F9, 1FA, 1FB, AND 1FC.
                ;2839 ;
                ;2840 ; SUBTEST 4 - CURRENT CONTEXT = QUAD. CHECK THE D REGISTER BYTE SELECT
                ;2841 ;      ON A SECOND REFERANCE AND A VA OF 3.
                ;2842 ;
                ;2843 ; LOGIC DESCRIPTION
                ;2844 ;
                ;2845 ; THIS TEST CHECKS THE SAVED CONTEXT LATCH, THE MEMORY REFERANCE
                ;2846 ; DATA TYPE MUX, AND THE UNALIGNED AND PAGE TRAP MUX ON THE CEH MODULE.
                ;2847 ;
                ;2848 ; ERROR DESCRIPTION
                ;2849 ;
                ;2850 ; SUBTESTS 1, 2, AND 3
                ;2851 ;  DATA: EXPECTED MICRO TRAP FLAGS
                ;2852 ;        RECEIVED MICRO TRAP FLAGS
                ;2853 ;        LOOP COUNT - INDICATES WHICH VALUE IS CURRENTLY IN THE VA
                ;2854 ; SUBTEST 4
                ;2855 ;  DATA: EXPECTED CONTENTS OF THE D REGISTER
                ;2856 ;        RECEIVED CONTENTS OF THE D REGISTER
                ;2857 ;--
                ;2858 ;*****************************************************************************
                ;2859
                ;2860 =0
U 10EC, 0000.003D,0180.0800,0000.105F  ;2861 T006: NEWTST
U 10ED, 0018.0038,0D80.09F8,0000.1296  ;2862  RC[0F]_K[.3]
U 1296, 0818.0038,85F8.0800,0084.7297  ;2863  D_K[.C],SC_K[.C],Q_0
U 1297, 0D00.003C,75F0.2C00,0000.1298  ;2864  D_DAL.SC,Q_ID[MAINT]
U 1298, 081D.0030,0180.0800,0000.1299  ;2865  D_D.OR.Q
U 1299, 0000.003C,7580.3C00,0000.10EE  ;2866  ID[MAINT]_D
                ;2867
                ;2868 ;/////////////////////////////////////////////////////////////////////////////
                ;2869 ;*
                ;2870 ; CHECK THE UNALIGNED TRAP WITH CURRENT CONTEXT = QUAD
                ;2871 ;-
                ;2872
U 10EE, 0000.003D,0180.0800,0000.1196  ;2873 =0 SUBTEST
U 10EF, 0018.0038,D580.09F0,0000.10F0  ;2874  RC[0E]_K[.6]   ; SET THE EXPECTED TRAP FLAG
```

```
U 10F0. 2018.0039.9D80.0980.4200.119C  ;2875 =0 RC[0]_K[.7C],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 10F1. 0003.003C.0180.0988.0000.10F2  ;2876  RC[1]_0    ; SET THE INITIAL VA VALUE
U 10F2. 0000.003D.0180.0800.0000.119B  ;2877 =0 CALL,J/DISPAT    ; LATCH THE DATA TYPE
U 10F3. 0018.0038.0D80.09E0.0000.10F4  ;2878  RC[0C]_K[.3]    ; SET THE LOOP COUNT
U 10F4. 0000.003D.0180.0800.0000.1191  ;2879 =0 ERLOOP
U 10F5. 0018.0038.D580.0AF8.0000.129A  ;2880 LOOPA: R[0F]_K[.6]    ; SET THE TRAP FLAG
U 129A. 0010.0038.0180.0908.0200.129B  ;2881  ALU_RC[1],VA_ALU   ; SET THE VA
U 129B. 0000.C03C.0180.4000.0000.129C  ;2882  MCT/READ.V.RCHK,DT/INST.DEP ; EXECUTE THE MEMORY FUNCTION
U 129C. 0000.003C.01C0.0A78.0000.129D  ;2883  Q_R[0F]     ; GET THE TRAP FLAG
U 129D. 0810.0038.0180.0970.0000.129E  ;2884  D_RC[0E]    ; GET THE EXPECTED TRAP FLAG
U 129E. 001D.2000.0180.0800.0010.129F  ;2885  ALU_Q-D,CLK.UBCC   ; CHECK IT
U 129F. 0000.013C.0180.0800.0000.10F6  ;2886  Z?
U 10F6. 0001.203D.0180.09E8.0000.1194  ;2887 =0 ERROR2,RC[0D]_Q    ; TRAP FLAG INCORRECT
        ;2888
U 10F7. 0810.0038.0180.0908.0000.12A0  ;2889  D_RC[1]     ; GET VA VALUE
U 12A0. 0019.0014.0580.0988.0000.12A1  ;2890  RC[1]_D+K[.1]    ; SELECT NEXT VA VALUE
U 12A1. 0018.0038.1180.09F0.0000.12A2  ;2891  RC[0E]_K[.4]    ; SET THE EXPECTED TRAP FLAG
U 12A2. 0810.0038.0180.0960.0000.12A3  ;2892  D_RC[0C]    ; GET THE LOOP COUNT
U 12A3. 0019.0000.0580.09E0.0010.12A4  ;2893  RC[0C]_D-K[.1],CLK.UBCC   ; DECREMENT AND CHECK
U 12A4. 0000.013C.0180.0800.0000.10FB  ;2894  Z?
U 10F8. 0000.003C.0180.0800.0000.10F5  ;2895 =0 J/LOOPA
U 10F9. 0000.003C.0180.0800.0000.10FA  ;2896  NOP
        ;2897
        ;2898 ;////////////////////////////////////////////////////////////////////////
        ;2899 ;+
        ;2900 ; CHECK THE UNALIGNED TRAP WITH SAVED CONTEXT = QUAD
        ;2901 ;-
        ;2902
U 10FA. 0000.003D.0180.0800.0000.1196  ;2903 =0 SUBTEST
U 10FB. 0018.0038.1180.09F0.0000.10FC  ;2904  RC[0E]_K[.4]    ; SET THE EXPECTED TRAP FLAG
U 10FC. 0000.003D.0980.0800.0084.7198  ;2905 =0 SETRCF[.2]
U 10FD. 0810.0038.0180.0900.0000.12A5  ;2906  D_RC[0]
U 12A5. 0019.0014.1180.0990.0000.10FE  ;2907  RC[2]_D+K[.4]    ; SET THE IB ADDRESS
U 10FE. 0000.003D.0180.0800.0000.1191  ;2908 =0 ERLOOP
U 10FF. 0018.0038.D580.0AF8.0000.110A  ;2909  R[0F]_K[.6]    ; SET THE TRAP FLAG
U 110A. 2010.0039.0180.0900.4200.119C  ;2910 =0 ALU_RC[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 110B. 0000.003C.0180.0800.0000.1110  ;2911  NOP     ; WITH A QUAD INSTRUCTION
U 1110. 0000.003D.0180.0800.0000.119B  ;2912 =0 CALL,J/DISPAT    ; LATCH THE DATA TYPE
U 1111. 0003.003C.0180.0800.0200.12A6  ;2913  VA_ALU,ALU_0(A)    ; LOAD THE VA
U 12A6. 0000.C03C.0180.4000.0000.1112  ;2914  MCT/READ.V.RCHK,DT/INST.DEP ; LOAD THE SAVED CONTEXT LATCH
U 1112. 2010.0039.0180.0910.4200.119C  ;2915 =0 ALU_RC[2],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB WITH A WORD INSTR
U 1113. 0000.003C.0180.0800.0000.1114  ;2916  NOP     ; TO CHECK THAT THE CONTEXT MUX
        ;2917       ; SWITCHES TO THE SAVED CONTEXT INPUTS
U 1114. 0000.003D.0180.0800.0000.119B  ;2918 =0 CALL,J/DISPAT    ; LATCH THE WORD DATA TYPE
U 1115. 0018.0038.0580.0800.0200.12A7  ;2919  VA_ALU,ALU_K[.1]   ; SETUP THE VA
U 12A7. 0000.C03C.0180.4000.3800.12A8  ;2920  MCT/READ.V.RCHK,DT/INST.DEP,MSC/RETRY.TRAP ; THIS SHOULD TRAP
U 12A8. 0000.003C.01C0.0A78.0000.12A9  ;2921  Q_R[0F]     ; GET THE TRAP FLAG
U 12A9. 0810.0038.0180.0970.0000.12AB  ;2922  D_RC[0E]    ; AND THE EXPECTED TRAP FLAG
U 12AB. 001D.2000.0180.0800.0010.12AC  ;2923  ALU_Q-D,CLK.UBCC   ; CHECK IT
U 12AC. 0000.013C.0180.0800.0000.1116  ;2924  Z?
U 1116. 0001.203D.0180.09E8.0000.1194  ;2925 =0 ERROR2,RC[0D]_Q    ; TRAP DID NOT OCCUR
U 1117. 0000.003C.0180.0800.0000.1118  ;2926  NOP
        ;2927
        ;2928 ;////////////////////////////////////////////////////////////////////////
        ;2929 ;+
```

```
                ;2930 ; CHECK THE PAGE TRAP AND CURRENT CONTEXT = QUAD
                ;2931 ;-
                ;2932
U 1118. 0000.003D.0180.0800.0000.1196  ;2933 =0 SUBTEST
U 1119. 0018.0038.9D80.0980.0000.12AD  ;2934  RC[0]_K[.7C]    ; SET THE IB ADDRESS
U 12AD. 0018.0038.D580.09F0.0000.111A  ;2935  RC[0E]_K[.6]    ; SET THE EXPECTED TRAP FLAG
U 111A. 0000.003D.0D80.0800.0084.7198  ;2936 =0 SETRCF[.3]
U 111B. 0818.0038.1180.0800.0000.12AE  ;2937  D_K[.4]
U 12AE. 0019.0014.0580.09E0.0000.111C  ;2938  RC[0C]_D+K[.1]    ; SET THE LOOP COUNT
U 111C. 2010.0039.0180.0900.4200.119C  ;2939 =0 ALU_RC[0],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 111D. 0000.003C.0180.0800.0000.111E  ;2940  NOP
U 111E. 0000.003D.0180.0800.0000.119B  ;2941 =0 CALL,J/DISPAT    ; LATCH THE DATA TYPE
U 111F. 0000.003C.0180.0800.0000.1120  ;2942  NOP
U 1120. 0818.0039.8D80.0800.0000.11F3  ;2943 =0 D_K[.1F],CALL,J/DC8   ; GENERATE INITIAL VA VALUE
U 1121. 0001.003C.0180.0988.0200.12AF  ;2944  RC[1]_D,VA_ALU    ; SAVE AND LOAD THE VA
U 12AF. 0000.003C.2180.0800.0084.72B0  ;2945  SC_K[.14]    ; TURN THE TB ON
U 12B0. 0803.001C.0180.0800.0000.12B1  ;2946  D_NOT.MASK
U 12B1. 0819.0030.0580.0800.0000.12B2  ;2947  D_D.OR.K[.1]
U 12B2. 0000.003C.4980.3C00.0000.12B3  ;2948  ID[TBER0]_D    ;...
U 12B3. 0818.0038.CD80.0800.0000.12B4  ;2949  D_K[.F0]    ; GENERATE A VALID PTE
U 12B4. 0819.0014.0180.0800.0000.12B5  ;2950  D_D+K[.8]
U 12B5. 0B00.003C.0180.0800.0000.12B6  ;2951  D_D.SWAP
U 12B6. 0000.003C.4180.3C00.0000.12B7  ;2952  ID[TBUF]_D    ; LOAD THE TB
U 12B7. 0000.003C.0180.0803.0000.12B8  ;2953  VA_VA+4
U 12B8. 0000.003C.4180.3C00.0000.1122  ;2954  ID[TBUF]_D
U 1122. 0000.003D.0180.0800.0000.1191  ;2955 =0 ERLOOP
U 1123. 0018.0038.D580.0AF8.0000.12B9  ;2956 LOOPB: R[0F]_K[.6]    ; SET THE TRAP FLAG
U 12B9. 0010.0038.0180.0908.0200.12BA  ;2957  ALU_RC[1],VA_ALU   ; LOAD THE VA
U 12BA. 0000.C03C.0180.4000.0000.12BB  ;2958  MCT/READ.V.RCHK,DT/INST.DEP ; EXECUTE THE MEMORY FUNCTION
U 12BB. 0000.003C.01C0.0A78.0000.12BC  ;2959  Q_R[0F]    ; GET THE TRAP FLAG
U 12BC. 0810.0038.0180.0970.0000.12BD  ;2960  D_RC[0E]    ; AND THE EXPECTED FLAG
U 12BD. 001D.2000.0180.0800.0010.12BE  ;2961  ALU_Q-D,CLK.UBCC   ; CHECK IT
U 12BE. 0000.013C.0180.0800.0000.1124  ;2962  Z?
U 1124. 0001.203D.0180.09E8.0000.1194  ;2963 =0 ERROR2,RC[0D]_Q    ; TRAP FLAG INCORRECT
                ;2964
U 1125. 0810.0038.0180.0908.0000.12BF  ;2965  D_RC[1]    ; GET CURRENT VA VALUE
U 12BF. 0019.0014.0580.0988.0000.12C0  ;2966  RC[1]_D+K[.1]    ; INCREMENT IT
U 12C0. 0018.0038.0980.09F0.0000.12C1  ;2967  RC[0E]_K[.2]    ; SET THE EXPECTED DATA
U 12C1. 0810.0038.0180.0960.0000.12C2  ;2968  D_RC[0C]    ; GET THE LOOP COUNT
U 12C2. 0019.0000.0580.09E0.0010.12C3  ;2969  RC[0C]_D-K[.1],CLK.UBCC   ; DECREMENT AND CHECK
U 12C3. 0000.013C.0180.0800.0000.1126  ;2970  Z?
U 1126. 0000.003C.0180.0800.0000.1123  ;2971 =0 J/LOOPB
U 1127. 0000.003C.0180.0800.0000.1128  ;2972  NOP
                ;2973
                ;2974 ;////////////////////////////////////////////////////////////////////////
                ;2975 ;*
                ;2976 ; CHECK THAT THE D REGISTER GETS THE CORRECT DATA FOR QUAD DATA TYPE
                ;2977 ;-
                ;2978
U 1128. 0000.003D.0180.0800.0000.1196  ;2979 =0 SUBTEST
U 1129. 0000.003C.0180.0800.0000.112A  ;2980  NOP
U 112A. 0000.003D.0980.0800.0084.7198  ;2981 =0 SETRCF[.2]
U 112B. 0F00.003C.0180.0800.0000.12C4  ;2982  D_0
U 12C4. 0000.003C.4980.3C00.0000.12C5  ;2983  ID[TBER0]_D    ; TURN OFF THE TB
U 12C5. 0018.0038.4180.0800.0200.12C6  ;2984  VA_ALU,ALU_K[.80]
```

```
U 12C6. 0000.003C.01FB.0800.0084.72C7  ;2985  SC_K[.8],Q_0
U 12C7. 0818.0038.C180.0800.0000.12C8  ;2986  D_K[.FFFF]   ; GET DATA PATTERN TO WRITE
U 12C8. 0D00.003C.0180.0800.0000.12C9  ;2987  D_DAL.SC
U 12C9. 0000.003C.0180.9800.0000.12CA  ;2988  MCT/VALIDATE   ; WRITE THE DATA PATTERN
U 12CA. 0D00.003C.0180.0800.0000.12CB  ;2989  D_DAL.SC
U 12CB. 0001.003C.0180.09F0.0000.112C  ;2990  RC[0E]_D   ; SAVE
U 112C. 0000.003D.0180.0800.0000.1191  ;2991 =0 ERLOOP
U 112D. 0003.003C.0180.0800.0200.12CC  ;2992  VA_ALU,ALU_0(A)   ; LOAD THE VA
U 12CC. 0000.C03C.0180.4000.0000.12CD  ;2993  MCT/READ.V.RCHK,DT/INST.DEP ; LATCH THE SAVED CONTEXT
U 12CD. 0818.0038.4180.0800.0000.12CE  ;2994  D_K[.80]
U 12CE. 0F19.0014.0D80.0800.0200.12CF  ;2995  VA_ALU,ALU_D+K[.3],D_0  ; LOAD THE VA
U 12CF. 0000.C03C.0180.4000.3000.12D0  ;2996  MCT/READ.V.RCHK,DT/INST.DEP,MSC/SECOND.REF ; EXECUTE THE TEST
U 12D0. 0010.0038.01C0.0970.0000.12D1  ;2997  Q_RC[0E]   ; GET THE EXPECTED DATA
U 12D1. 001D.2000.0180.0800.0010.12D2  ;2998  ALU_Q-D,CLK.UBCC  ; CHECK THE DATA
U 12D2. 0000.013C.0180.0800.0000.112E  ;2999  Z?
U 112E. 0001.003D.0180.09E8.0000.1194  ;3000 =0 ERROR2,RC[0D]_D   ; DATA IN WRONG POSITION
U 112F. 0000.003C.0180.0800.0000.1130  ;3001  NOP
            ;3002
            ;3003
            ;3004 ; FOLLOWING ARE THE IB OPCODES FOR THE ABOVE TEST:
            ;3005 ;% 7C
            ;3006 ;$ 6070,0000
            ;3007 ;$ 60A1,0000
            ;3008
```

```
                ;3009 .PAGE "TEST 174 ALU Z BRANCH WITH DATA TYPE QUAD"
                ;3010 ;*****************************************************************
                ;3011 ;**
                ;3012 ;
                ;3013 ; TEST 174 ALU Z BRANCH WITH DATA TYPE QUAD
                ;3014 ;
                ;3015 ; TEST DESCRIPTION
                ;3016 ;
                ;3017 ; THIS TEST CHECKS THE ALU = ZERO BRANCH FOR A FLOATING
                ;3018 ; POINT INSTRUCTION DATA TYPE.
                ;3019 ;
                ;3020 ; SUBTEST 1 - CHECK BITS<15:07>=0 MAKES THE BRANCH TRUE
                ;3021 ;
                ;3022 ; SUBTEST 2 - CHECK BITS<15:08>=0 MAKE THE BRANCH FALSE
                ;3023 ;
                ;3024 ; SUBTEST 3 - CHECK BITS<15:08>.NE.0 MAKE THE BRANCH FALSE
                ;3025 ;
                ;3026 ; LOGIC DESCRIPTION
                ;3027 ;
                ;3028 ; THIS TEST CHECKS THE ALU=0 SIGNAL ON THE CEH MODULE FOR FLOATING
                ;3029 ; DATA TYPE.
                ;3030 ;
                ;3031 ; ERROR DESCRIPTION
                ;3032 ;
                ;3033 ; NONE. WILL ONLY OCCUR IF THE BRANCH FAILS.
                ;3034 ;--
                ;3035 ;*****************************************************************
                ;3036
                ;3037 =0
U 1130. 0000.003D.0180.0800.0000.105F  ;3038 T007: NEWTST
U 1131. 0018.0038.1980.09F8.0000.1132  ;3039  RC[0F]_K[ZERO]
U 1132. 2018.0039.9D80.0800.4200.119C  ;3040 =0 ALU_K[.7C],FLUSH.IB,CALL,J/IBWAIT ; LOAD THE IB
U 1133. 0000.003C.0180.0800.0000.1134  ;3041  NOP
U 1134. 0000.003D.0180.0800.0000.119B  ;3042 =0 CALL,J/DISPAT   ; LATCH THE DATA TYPE
                ;3043
                ;3044 ;/////////////////////////////////////////////////////////////////
                ;3045 ;*
                ;3046 ; CHECK THE BRANCH TRUE WHEN BITS <15:07> = 0
                ;3047 ;-
                ;3048
U 1135. 0000.003C.0180.0800.0000.1136  ;3049  NOP
U 1136. 0000.003D.0180.0800.0000.1196  ;3050 =0 SUBTEST
U 1137. 0818.0038.C180.0800.0000.12D3  ;3051  D_K[.FFFF]   ; GENERATE THE TEST PATTERN
U 12D3. 0B00.003C.F980.0800.0000.12D4  ;3052  D_D.SWAP,K[.7E]
U 12D4. 0819.0014.F980.0800.0000.12D5  ;3053  D_D+K[.7E]
U 12D5. 0019.0014.0580.09F0.0000.1138  ;3054  RC[0E]_D+K[.1]   ; EQUALS FFFF007F
U 1138. 0000.003D.0180.0800.0000.1191  ;3055 =0 ERLOOP
U 1139. 0010.C038.0180.0970.0010.12D6  ;3056  ALU_RC[0E],CLK.UBCC,DT/INST.DEP ; EXECUTE THE TEST
U 12D6. 0000.013C.0180.0800.0000.113A  ;3057  Z?
U 113A. 0000.003D.0180.0800.0000.1194  ;3058 =0 ERROR2    ; SHOULD HAVE BEEN ZERO
U 113B. 0000.003C.0180.0800.0000.113C  ;3059  NOP
                ;3060
                ;3061 ;/////////////////////////////////////////////////////////////////
                ;3062 ;*
                ;3063 ; CHECK THE BRANCH WHEN BITS <15:08> = 0 AND BIT 7 = 1
```

```
            ;3064 ;-
            ;3065
U 113C, 0000.003D.0180.0800.0000.1196  ;3066 =0 SUBTEST
U 113D, 0810.0038.0180.0970.0000.12D7  ;3067  D_RC[0E]    ; GET LAST PATTERN
U 12D7, 0019.0030.4180.09F0.0000.113E  ;3068  RC[0E]_D.OR.K[.80]  ; SET BIT 7
U 113E, 0000.003D.0180.0800.0000.1191  ;3069 =0 ERLOOP
U 113F, 0010.C038.0180.0970.0010.12D8  ;3070  ALU_RC[0E],CLK.UBCC,DT/INST.DEP ; EXECUTE THE TEST
U 12D8, 0000.013C.0180.0800.0000.1140  ;3071  Z?
U 1140, 0000.003C.0180.0800.0000.1144  ;3072 =0 J/T007S3    ;
U 1141, 0000.003C.0180.0800.0000.1142  ;3073  NOP
U 1142, 0000.003D.0180.0800.0000.1194  ;3074 =0 ERROR2     ; DID NOT DETECT NON ZERO
U 1143, 0000.003C.0180.0800.0000.1144  ;3075  NOP
            ;3076
            ;3077 ;////////////////////////////////////////////////////////////////////////
            ;3078 ;*
            ;3079 ; CHECK THE BRANCH WHEN BITS <15:08> .NE. 0 AND BIT 7 = 0
            ;3080 ;-
            ;3081
            ;3082 =0
U 1144, 0000.003D.0180.0800.0000.1196  ;3083 T007S3: SUBTEST
U 1145, 081B.0000.0580.0800.0000.12D9  ;3084  D_0-K[.1]    ; GENERATE THE DATA PATTER
U 12D9, 0819.0024.4980.0800.0000.12DA  ;3085  D_D.ANDNOT.K[.FF]
U 12DA, 0819.0030.F980.0800.0000.12DC  ;3086  D_D.OR.K[.7E]
U 12DC, 0019.0014.0580.09F0.0000.1146  ;3087  RC[0E]_D*K[.1]
U 1146, 0000.003D.0180.0800.0000.1191  ;3088 =0 ERLOOP
U 1147, 0010.C038.0180.0970.0010.12DE  ;3089  ALU_RC[0E],CLK.UBCC,DT/INST.DEP ; EXECUTE THE TEST
U 12DE, 0000.013C.0180.0800.0000.1148  ;3090  Z?
U 1148, 0000.003C.0180.0800.0000.114B  ;3091 =0 J/T007END
U 1149, 0000.003C.0180.0800.0000.114A  ;3092  NOP
U 114A, 0000.003D.0180.0800.0000.1194  ;3093 =0 ERROR2     ; BRANCH FAILED
            ;3094 T007END:
U 114B, 0000.003C.0180.0800.0000.114C  ;3095  NOP
            ;3096
```

```
                ;3097 .PAGE "TEST 175 READ.V.NEWPC, WITH TB MISS, TEST"
                ;3098 ;*****************************************************************
                ;3099 ;++
                ;3100 ; TEST 175 READ.V.NEWPC WITH TB MISS TEST
                ;3101 ;
                ;3102 ; TEST DESCRIPTION
                ;3103 ;  THIS IS A TEST OF A READ.V.NEWPC WHICH RESULTS
                ;3104 ;  IN A TB MISS WITH MME=1.
                ;3105 ;  AFTER THIS ERROR CONDITION THE CONTENTS OF TBER1
                ;3106 ;  IS CHECKED AND A CALL 3 IS EXECUTED TO TEST THE
                ;3107 ;  IBUF'S REACTION TO THE ERROR.
                ;3108 ;
                ;3109 ; LOGIC DESCRIPTION
                ;3110 ;
                ;3111 ; ERROR DESCRIPTION
                ;3112 ; VA REFERENCED
                ;3113 ; EXPECTED TBER1
                ;3114 ; ACTUAL TBER1
                ;3115 ; ERROR CODE: (FOR FAILCHAIN)
                ;3116 ;  0=> TBER1 FAILED
                ;3117 ;  1=> IRC UPC-DECODE FAILED (WITH TB MISS ERROR)
                ;3118 ;  4=> BEN/B FAILED (WITH TB MISS ERROR)
                ;3119 ;  3=> IRC UPC-DECODE FAILED (WITHOUT TB MISS ERROR)
                ;3120 ;  2=> BEN/B FAILED (WITHOUT TB MISS ERROR)
                ;3121 ;  BITS <1:0> = EXPECTED <TBMX IB MISS L : TBMX IB ERR L> RESPECTIVELY
                ;3122 ;
                ;3123 ; SYNC POINT DESCRIPTION
                ;3124 ;
                ;3125 ;--
                ;3126 ;*****************************************************************
                ;3127 =0
U 114C. 0000.003D.0180.0800.0000.105F   ;3128 IBER0: NEWTST
U 114D. 0F18.0038.1180.09F8.0000.114E   ;3129  D_0,RC[0F]_K[.4]
U 114E. 0000.003D.4980.3C00.0000.11AD   ;3130 =0 ID[TBER0]_D,CALL,J/CSOFF
U 114F. 2000.003C.0180.0800.0000.1150   ;3131  IBC/FLUSH
U 1150. 0000.003D.0180.0800.0000.1207   ;3132 =0 CALL,J/TBWPTP
U 1151. 0F00.003C.0180.0800.0000.12DF   ;3133  D_0
U 12DF. 0000.003C.4D80.3C00.0000.12E0   ;3134  ID[TBER1]_D
U 12E0. 0003.003C.0180.0AF8.0000.12E1   ;3135  R[0F]_0
U 12E1. 0018.0038.1980.09F0.0200.12E2   ;3136  VA_K[ZERO],RC[0E]_K[ZERO]
U 12E2. 0000.003C.0180.6000.0000.12E3   ;3137  FS/MCT,MCT/READ.V.NEWPC
U 12E3. 0000.003C.4DF0.2C00.0000.12E4   ;3138  Q_ID[TBER1]
U 12E4. 0003.003C.0180.09E8.0000.12E5   ;3139  RC[0D]_0
U 12E5. 0001.203C.0180.09E0.0000.12E6   ;3140  RC[0C]_Q
U 12E6. 0019.2024.3180.0800.0010.12E7   ;3141  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 12E7. 0003.013C.0180.09D8.0000.1152   ;3142  Z?,RC[0B]_0    ; SET ERROR CODE
U 1152. 0000.003D.0180.0800.0000.1192   ;3143 =0 ERROR1     ; TBER1 ERROR BITS
                ;3144       ; NOT CLEAR.
U 1153. 0010.0038.0180.0908.0200.12E8   ;3145  VA_RC[1]
U 12E8. 0000.003C.0180.0900.0000.1156   ;3146  LC_RC[0]
U 1156. 0010.0039.0180.0AC0.0000.1032   ;3147 =0 R[8]_LC,CALL,J/TBWRB
U 1157. 0010.0038.0180.0920.0200.12E9   ;3148  VA_RC[4]
U 12E9. 0010.0038.0180.09F0.0000.12EA   ;3149  RC[0E]_LC
U 12EA. 0000.003C.0180.6000.0000.12EB   ;3150  FS/MCT,MCT/READ.V.NEWPC
U 12EB. 0000.003C.4DF0.2C00.0000.12EC   ;3151  Q_ID[TBER1]
```

```
U 12EC. 0001.203C.0180.09E0.0000.12ED  ;3152  RC[0C]_Q
U 12ED. 0019.2024.31C0.0800.0000.12EE  ;3153  ALU_Q[ANDNOT]K[.40],Q_ALU
U 12EE. 0818.0038.0180.0800.0000.12EF  ;3154  D_K[.8]
U 12EF. 0001.003C.0180.09E8.0000.12F0  ;3155  RC[0D]_D
U 12F0. 001D.2000.0180.0800.0010.12F1  ;3156  ALU_Q-D,CLK.UBCC
U 12F1. 0000.013C.0180.0800.0000.1158  ;3157  Z?
U 1158. 0000.003D.0180.0800.0000.1192  ;3158 =0 ERROR1    ; TBER1 NOT SET CORRECTLY.
U 1159. 0018.0038.0580.09D8.0000.1028  ;3159  RC[0B]_K[.1]    ; SET ERROR CODE
U 1028. 0000.003D.0180.0800.0000.1226  ;3160 =000 CALL,J/TARGC3
U 1029. 0000.003C.0180.0800.0000.12F2  ;3161  J/IBER0A  ; OK. INDICATION OF TB MISS.
U 102A. 0000.003D.0180.0800.0000.1192  ;3162  ERROR1    ; RESPONSE WAS A NON TB MISS TB ERROR.
U 102B. 0000.003D.0180.0800.0000.1192  ;3163  ERROR1    ; IBUF STALL BUT NO RESPONSE
          ;3164      ; TO TB MISS CONDITION.
U 102C. 0000.003D.0180.0800.0000.1192  ;3165  ERROR1    ; RESPONSE WAS TB MISS AND OTHER ERROR.
          ;3166 =
U 12F2. 0018.0B38.1180.09D8.0000.1034  ;3167 IBER0A: BEN/IB.TEST,RC[0B]_K[.4] ; SET ERROR CODE
U 1034. 0000.003C.0180.0800.0000.12F3  ;3168 =100 J/IBR0A
U 1035. 0000.003D.0180.0800.0000.1192  ;3169  ERROR1
U 1036. 0000.003D.0180.0800.0000.1192  ;3170  ERROR1
U 1037. 0000.003D.0180.0800.0000.1192  ;3171  ERROR1
U 12F3. 0010.0038.0180.0908.0200.12F4  ;3172 IBR0A: VA_RC[1]
U 12F4. 0010.0038.0180.09F0.0000.12F5  ;3173  RC[0E]_LC
U 12F5. 0000.003C.0180.6000.0000.12F6  ;3174  FS/MCT,MCT/READ.V.NEWPC
U 12F6. 0000.003C.4DF0.2C00.0000.12F7  ;3175  Q_ID[TBER1]
U 12F7. 0003.003C.0180.09E8.0000.12F8  ;3176  RC[0D]_Q
U 12F8. 0001.203C.0180.09E0.0000.12F9  ;3177  RC[0C]_Q
U 12F9. 0019.2024.3180.0800.0010.12FA  ;3178  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 12FA. 0003.013C.0180.09D8.0000.115A  ;3179  Z?,RC[0B]_Q    ; SET ERROR CODE
U 115A. 0000.003D.0180.0800.0000.1192  ;3180 =0 ERROR1    ; CAN'T CLEAR THE ERROR CONDITION.
U 115B. 0018.0038.0D80.09D8.0000.1038  ;3181  RC[0B]_K[.3]    ; SET ERROR CODE
U 1038. 0000.003D.0180.0800.0000.1226  ;3182 =000 CALL,J/TARGC3
U 1039. 0000.003D.0180.0800.0000.1192  ;3183  ERROR1
U 103A. 0000.003D.0180.0800.0000.1192  ;3184  ERROR1
U 103B. 0000.003C.0180.0800.0000.12FB  ;3185  J/IBER0B
U 103C. 0000.003D.0180.0800.0000.1192  ;3186  ERROR1
          ;3187 =
U 12FB. 0018.0B38.0980.09D8.0000.1044  ;3188 IBER0B: BEN/IB.TEST,RC[0B]_K[.2] ; SET ERROR CODE
U 1044. 0000.003D.0180.0800.0000.1192  ;3189 =100 ERROR1
U 1045. 0000.003D.0180.0800.0000.1192  ;3190  ERROR1
U 1046. 0000.003C.0180.0800.0000.12FC  ;3191  J/IBR0B
U 1047. 0000.003D.0180.0800.0000.1192  ;3192  ERROR1
U 12FC. 0000.003C.0180.0800.0000.115C  ;3193 IBR0B: NOP
          ;3194
```

```
                ;3195 .PAGE "TEST 176 READ.V.NEWPC, WITH ACCESS VIOLATION, TEST"
                ;3196 ;*****************************************************************
                ;3197 ;++
                ;3198 ; TEST 176 READ.V.NEWPC, WITH ACCESS VIOLATION, TEST
                ;3199 ;
                ;3200 ; TEST DESCRIPTION
                ;3201 ;  THIS IS A TEST OF A READ.V.NEWPC WHICH RESULTS IN A TB
                ;3202 ;  HIT THAT IS ACTUALLY AN ACCESS VIOLATION. THE CONTENTS
                ;3203 ;  OF TBER1 IS CHECKED AND A CALL 3 IS EXECUTED TO TEST
                ;3204 ;  THE IBUF'S RESPONSE TO THIS ERROR CONDITION.
                ;3205 ;
                ;3206 ; LOGIC DESCRIPTION
                ;3207 ;
                ;3208 ; ERROR DESCRIPTION
                ;3209 ; VA
                ;3210 ; PTE
                ;3211 ; EXPECTED TBER1
                ;3212 ; GOT TBER1
                ;3213 ; ERROR CODE: (FOR FAILCHAIN)
                ;3214 ;  0=> TBER1 FAILED
                ;3215 ;  2=> IRC UPC-DECODE FAILED (WITH ACCESS VIOLATION)
                ;3216 ;  1=> BEN/B FAILED (WITH ACCESS VIOLATION)
                ;3217 ;  3=> IRC UPC-DECODE FAILED (WITHOUT ACCESS VIOLATION)
                ;3218 ;  6=> BEN/B FAILED (WITHOUT ACCESS VIOLATION)
                ;3219 ;  BITS <1:0> = EXPECTED <TBMX IB MISS L : TBMX IB ERR L> RESPECTIVELY
                ;3220 ;
                ;3221 ; SYNC POINT DESCRIPTION
                ;3222 ;
                ;3223 ;--
                ;3224 ;*****************************************************************
                ;3225 =0
U 115C. 0000.003D.0180.0800.0000.105F   ;3226 IBER1: NEWTST
U 115D. 0F1B.0010.1180.09F8.0000.115E   ;3227  D_0,ALU_0+K[.4]+1,RC[0F]_ALU
U 115E. 0000.003D.4980.3C00.0000.11AD   ;3228 =0 ID[TBER0]_D,CALL,J/CSOFF
U 115F. 2000.003C.0180.0800.0000.1160   ;3229  IBC/FLUSH
U 1160. 0000.003D.0180.0800.0000.1207   ;3230 =0 CALL,J/TBWPTP
U 1161. 0F00.003C.0180.0800.0000.12FD   ;3231  D_0
U 12FD. 0000.003C.4D80.3C00.0000.12FE   ;3232  ID[TBER1]_D
U 12FE. 0018.0038.1980.09F0.0200.12FF   ;3233  VA_K[ZERO],RC[0E]_K[ZERO]
U 12FF. 0000.003C.0180.6000.0000.1300   ;3234  FS/MCT,MCT/READ.V.NEWPC
U 1300. 0000.003C.4DF0.2C00.0000.1301   ;3235  Q_ID[TBER1]
U 1301. 0003.003C.0180.09E0.0000.1302   ;3236  RC[0C]_0
U 1302. 0001.203C.0180.09D8.0000.1303   ;3237  RC[0B]_Q
U 1303. 0019.2024.3180.0800.0010.1304   ;3238  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 1304. 0003.013C.0180.09D0.0000.1162   ;3239  Z?,RC[0A]_0    ; SET ERROR CODE
U 1162. 0000.003D.0180.0800.0000.1192   ;3240 =0 ERROR1
U 1163. 0010.0038.0180.0908.0200.1305   ;3241  VA_RC[1]
U 1305. 0010.0038.0180.09F0.0000.1306   ;3242  RC[0E]_LC
U 1306. 0000.003C.8D80.0800.0084.7307   ;3243  SC_K[.1F]
U 1307. 0003.001C.0180.09E8.0000.1164   ;3244  ALU_NOT.MASK,RC[0D]_ALU
U 1164. 0003.001D.0180.0AC0.0000.1032   ;3245 =0 ALU_NOT.MASK,R[8]_ALU,CALL,J/TBWRB
U 1165. 0000.003C.0180.6000.0000.1308   ;3246  FS/MCT,MCT/READ.V.NEWPC
U 1308. 0000.003C.4DF0.2C00.0000.1309   ;3247  Q_ID[TBER1]
U 1309. 0019.2024.31C0.0800.0000.130A   ;3248  ALU_Q[ANDNOT]K[.40],Q_ALU
U 130A. 0818.0038.0980.0800.0000.130B   ;3249  D_K[.2]
```

```
U 130B. 0001.003C.0180.09E0.0000.130C  ;3250  RC[0C]_D
U 130C. 0001.203C.0180.09D8.0000.130D  ;3251  RC[0B]_Q
U 130D. 001D.2000.0180.0800.0010.130E  ;3252  ALU_Q-D,CLK.UBCC
U 130E. 0000.013C.0180.0800.0000.1166  ;3253  Z?
U 1166. 0000.003D.0180.0800.0000.1192  ;3254 =0 ERROR1    ; TBER1 INCORRECT AFTER ACCESS VIOLATION.
U 1167. 0018.0038.0980.09D0.0000.1050  ;3255  RC[0A]_K[.2]    ; SET ERROR CODE
U 1050. 0000.003D.0180.0800.0000.1226  ;3256 =000 CALL,J/TARGC3
U 1051. 0000.003D.0180.0800.0000.1192  ;3257  ERROR1    ; ACKNOWLEDGED MISS INSTEAD OF ACCESS
          ;3258      ; VIOLATION.
U 1052. 0000.003C.0180.0800.0000.130F  ;3259  J/IBER1A   ; OK
U 1053. 0000.003D.0180.0800.0000.1192  ;3260  ERROR1    ; ERROR NOT ACKNOWLEDGED.
          ;3261      ; IB STALL.
U 1054. 0000.003D.0180.0800.0000.1192  ;3262  ERROR1    ; ACKNOWLEDGED MISS AND OTHER ERROR.
          ;3263 =
U 130F. 0018.0B38.0580.09D0.0000.1064  ;3264 IBER1A: BEN/IB.TEST,RC[0A]_K[.1] ; SET ERROR CODE
U 1064. 0000.003D.0180.0800.0000.1192  ;3265 =100 ERROR1
U 1065. 0000.003C.0180.0800.0000.1310  ;3266  J/IBR1A
U 1066. 0000.003D.0180.0800.0000.1192  ;3267  ERROR1
U 1067. 0000.003D.0180.0800.0000.1192  ;3268  ERROR1
U 1310. 0F00.003C.0180.0800.0000.1311  ;3269 IBR1A: D_0
U 1311. 0000.003C.4980.3C00.0000.1312  ;3270  ID[TBER0]_D
U 1312. 0000.003C.0180.6000.0000.1313  ;3271  FS/MCT,MCT/READ.V.NEWPC
U 1313. 0000.003C.4DF0.2C00.0000.1314  ;3272  Q_ID[TBER1]
U 1314. 0003.003C.0180.09E0.0000.1315  ;3273  RC[0C]_0
U 1315. 0001.203C.0180.09D8.0000.1316  ;3274  RC[0B]_Q
U 1316. 0019.2024.3180.0800.0010.1317  ;3275  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 1317. 0003.013C.0180.09D0.0000.1168  ;3276  Z?,RC[0A]_0    ; SET ERROR CODE
U 1168. 0000.003D.0180.0800.0000.1192  ;3277 =0 ERROR1    ; CAN'T CLEAR ERROR CONDITION IN TBER1
U 1169. 0018.0038.0D80.09D0.0000.1068  ;3278  RC[0A]_K[.3]    ; SET ERROR CODE
U 1068. 0000.003D.0180.0800.0000.1226  ;3279 =000 CALL,J/TARGC3
U 1069. 0000.003D.0180.0800.0000.1192  ;3280  ERROR1
U 106A. 0000.003D.0180.0800.0000.1192  ;3281  ERROR1
U 106B. 0000.003C.0180.0800.0000.1318  ;3282  J/IBER1B
U 106C. 0000.003D.0180.0800.0000.1192  ;3283  ERROR1
          ;3284 =
U 1318. 0018.0B38.D580.09D0.0000.1074  ;3285 IBER1B: BEN/IB.TEST,RC[0A]_K[.6] ; SET ERROR CODE
U 1074. 0000.003D.0180.0800.0000.1192  ;3286 =100 ERROR1
U 1075. 0000.003D.0180.0800.0000.1192  ;3287  ERROR1
U 1076. 0000.003C.0180.0800.0000.1319  ;3288  J/IBR1B
U 1077. 0000.003D.0180.0800.0000.1192  ;3289  ERROR1
U 1319. 0000.003C.0180.0800.0000.116A  ;3290 IBR1B: NOP
          ;3291
```

```
                ;3292 .PAGE "TEST 177 READ.V.NEWPC, WITH TB PARITY ERROR, TEST"
                ;3293 ;*****************************************************************
                ;3294 ;**
                ;3295 ; TEST 177 READ.V.NEWPC, WITH TB PARITY ERROR, TEST
                ;3296 ;
                ;3297 ; TEST DESCRIPTION
                ;3298 ;  THIS IS A TEST OF A READ.V.NEWPC WHICH RESULTS
                ;3299 ;  IN A TB PARITY ERROR. THE CONTENTS OF TBER1 IS CHECKED AND
                ;3300 ;  A CALL 3 IS EXECUTED TO TEST THE IBUF'S RESPONSE TO THIS ERROR
                ;3301 ;  CONDITION. NOTE THE PARITY ERROR USED IS FORCED IN GROUP 0
                ;3302 ;  BYTE 1.
                ;3303 ;
                ;3304 ; LOGIC DESCRIPTION
                ;3305 ;
                ;3306 ; ERROR DESCRIPTION
                ;3307 ; VA
                ;3308 ; EXPECTED TBER1
                ;3309 ; ACTUAL TBER1
                ;3310 ; ERROR CODE: (FOR FAILCHAIN)
                ;3311 ;  0=> TBER1 FAILED
                ;3312 ;  2=> IRC UPC-DECODE FAILED (WITH TB PARITY ERROR)
                ;3313 ;  1=> BEN/B FAILED (WITH TB PARITY ERROR)
                ;3314 ;  3=> IRC UPC-DECODE FAILED (WITHOUT TB PARITY ERROR)
                ;3315 ;  6=> BEN/B FAILED (WITHOUT TB PARITY ERROR)
                ;3316 ;  BITS <1:0> = EXPECTED <TBMX IB MISS L : TBMX IB ERR L> RESPECTIVELY
                ;3317 ;
                ;3318 ; SYNC POINT DESCRIPTION
                ;3319 ;
                ;3320 ;--
                ;3321 ;*****************************************************************
                ;3322 =0
U 116A. 0000.003D.0180.0800.0000.105F   ;3323 IBER2: NEWTST
U 116B. 0F18.0038.1180.09F8.0000.116C   ;3324  D_0,RC[0F]_K[.4]
U 116C. 0000.003D.4980.3C00.0000.11AD   ;3325 =0 ID[TBER0]_D,CALL,J/CSOFF
U 116D. 2000.003C.0180.0800.0000.116E   ;3326  IBC/FLUSH
U 116E. 0000.003D.0180.0800.0000.1207   ;3327 =0 CALL,J/TBWPTP
U 116F. 0F00.003C.0180.0800.0000.131A   ;3328  D_0
U 131A. 0000.003C.4D80.3C00.0000.131B   ;3329  ID[TBER1]_D
U 131B. 0018.0038.1980.09F0.0200.131C   ;3330  VA_K[ZERO],RC[0E]_K[ZERO]
U 131C. 0000.003C.0180.6000.0000.131D   ;3331  FS/MCT,MCT/READ.V.NEWPC
U 131D. 0000.003C.4DF0.2C00.0000.131E   ;3332  Q_ID[TBER1]
U 131E. 0003.003C.0180.09E8.0000.131F   ;3333  RC[0D]_0
U 131F. 0001.203C.0180.09E0.0000.1320   ;3334  RC[0C]_Q
U 1320. 0019.2024.3180.0800.0010.1321   ;3335  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 1321. 0003.013C.0180.09D8.0000.1170   ;3336  Z?,RC[0B]_0   ; SET ERROR CODE
U 1170. 0000.003D.0180.0800.0000.1192   ;3337 =0 ERROR1    ; TBER1 ERROR BITS NOT CLEARED.
U 1171. 0010.0038.0180.0908.0200.1322   ;3338  VA_RC[1]
U 1322. 0000.003C.0180.0900.0000.1172   ;3339  LC_RC[0]
U 1172. 0010.0039.0180.0AC0.0000.1032   ;3340 =0 R[8]_LC,CALL,J/TBWRB
U 1173. 0010.0038.0180.0920.0200.1323   ;3341  VA_RC[4]
U 1323. 0010.0038.0180.09F0.0000.1324   ;3342  RC[0E]_LC
U 1324. 0000.003C.0180.0918.0000.1174   ;3343  LC_RC[3]
U 1174. 0010.0039.0180.0AC0.0000.1030   ;3344 =0 R[8]_LC,CALL,J/TBWR0
U 1175. 0818.0038.5D80.0800.0000.1325   ;3345  D_K[.7]
U 1325. 0000.003C.4980.3C00.0000.1326   ;3346  ID[TBER0]_D
```

```
U 1326. 0000.003C.0180.6000.0000.1327  ;3347  FS/MCT,MCT/READ.V.NEWPC
U 1327. 0F00.003C.0180.0800.0000.1328  ;3348  D_0
U 1328. 0000.003C.4980.3C00.0000.1329  ;3349  ID[TBER0]_D
U 1329. 0000.003C.4DF0.2C00.0000.132A  ;3350  Q_ID[TBER1]
U 132A. 0001.003C.0180.09E8.0000.132B  ;3351  RC[0D]_D
U 132B. 0001.203C.0180.09E0.0000.132C  ;3352  RC[0C]_Q
U 132C. 0019.2024.31C0.0800.0000.132D  ;3353  ALU_Q[ANDNOT]K[.40],Q_ALU
U 132D. 0818.0038.1180.0800.0000.132E  ;3354  D_K[.4]
U 132E. 0000.003C.6580.0800.0084.732F  ;3355  SC_K[.10]
U 132F. 0801.001C.0180.09E8.0000.1330  ;3356  ALU_D.ORNOT.MASK,D_ALU,RC[0D]_ALU
U 1330. 001D.2000.0180.0800.0010.1331  ;3357  ALU_Q-D,CLK.UBCC
U 1331. 0000.013C.0180.0800.0000.1176  ;3358  Z?
U 1176. 0000.003D.0180.0800.0000.1192  ;3359 =0 ERROR1    ; TBER1 INCORRECT.
U 1177. 0018.0038.0980.09D8.0000.1078  ;3360  RC[0B]_K[.2]    ; SET ERROR CODE
U 1078. 0000.003D.0180.0800.0000.1226  ;3361 =000 CALL,J/TARGC3
U 1079. 0000.003D.0180.0800.0000.1192  ;3362  ERROR1    ; TB MISS ACKNOWLEDGED
U 107A. 0000.003C.0180.0800.0000.1332  ;3363  J/IBER2A
U 107B. 0000.003D.0180.0800.0000.1192  ;3364  ERROR1    ; PARITY ERROR NOT ACKNOWLEDGED.
U 107C. 0000.003D.0180.0800.0000.1192  ;3365  ERROR1
          ;3366 =
U 1332. 0018.0B38.0580.09D8.0000.1084  ;3367 IBER2A: BEN/IB.TEST,RC[0B]_K[.1] ; SET ERROR CODE
U 1084. 0000.003D.0180.0800.0000.1192  ;3368 =100 ERROR1
U 1085. 0000.003C.0180.0800.0000.1333  ;3369  J/IBR2A
U 1086. 0000.003D.0180.0800.0000.1192  ;3370  ERROR1
U 1087. 0000.003D.0180.0800.0000.1192  ;3371  ERROR1
U 1333. 0F00.003C.0180.0800.0000.1334  ;3372 IBR2A: D_0
U 1334. 0000.003C.4980.3C00.0000.1335  ;3373  ID[TBER0]_D
U 1335. 0000.003C.4D80.3C00.0000.1336  ;3374  ID[TBER1]_D
U 1336. 0000.003C.0180.6000.0000.1337  ;3375  FS/MCT,MCT/READ.V.NEWPC
U 1337. 0000.003C.4DF0.2C00.0000.1338  ;3376  Q_ID[TBER1]
U 1338. 0003.003C.0180.09E8.0000.1339  ;3377  RC[0D]_0
U 1339. 0001.203C.0180.09E0.0000.133A  ;3378  RC[0C]_Q
U 133A. 0019.2024.3180.0800.0010.133B  ;3379  ALU_Q[ANDNOT]K[.40],CLK.UBCC
U 133B. 0003.013C.0180.09D8.0000.1178  ;3380  Z?,RC[0B]_0    ; SET ERROR CODE
U 1178. 0000.003D.0180.0800.0000.1192  ;3381 =0 ERROR1    ; ERROR BITS IN TBER1 NOT CLEARED.
U 1179. 0018.0038.0D80.09D8.0000.1088  ;3382  RC[0B]_K[.3]    ; SET ERROR CODE
U 1088. 0000.003D.0180.0800.0000.1226  ;3383 =000 CALL,J/TARGC3
U 1089. 0000.003D.0180.0800.0000.1192  ;3384  ERROR1
U 108A. 0000.003D.0180.0800.0000.1192  ;3385  ERROR1
U 108B. 0000.003C.0180.0800.0000.133C  ;3386  J/IBER2B
U 108C. 0000.003D.0180.0800.0000.1192  ;3387  ERROR1
          ;3388 =
U 133C. 0018.0B38.D580.09D8.0000.1094  ;3389 IBER2B: BEN/IB.TEST,RC[0B]_K[.6] ; SET ERROR CODE
U 1094. 0000.003D.0180.0800.0000.1192  ;3390 =100 ERROR1
U 1095. 0000.003D.0180.0800.0000.1192  ;3391  ERROR1
U 1096. 0000.003C.0180.0800.0000.133D  ;3392  J/IBR2B
U 1097. 0000.003D.0180.0800.0000.1192  ;3393  ERROR1
U 133D. 0000.003C.0180.0800.0000.117A  ;3394 IBR2B: NOP
          ;3395
          ;3396
```

```
            ;3397   .PAGE "TEST 178 RESERVED"
            ;3398 =0
U 117A, 0000,003D,0180,0800,0000,105F  ;3399 T178: NEWTST
U 117B, 0000,003C,0180,0800,0000,1180  ;3400  NOP
            ;3401
```

```
            ;3402  .PAGE "TEST 179 RESERVED"
            ;3403 =0
U 1180, 0000,003D,0180,0800,0000,105F  ;3404 T179: NEWTST
U 1181, 0000,003C,0180,0800,0000,1182  ;3405  NOP
            ;3406
```

```
            ;3407  .PAGE "TEST 17A RESERVED"
            ;3408 =0
U 1182. 0000.003D.0180.0800.0000.105F  ;3409 T17A: NEWTST
U 1183. 0000.003C.0180.0800.0000.133E  ;3410  NOP
            ;3411
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 0056
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09          Page  10

            ;3412  .PAGE "DUMMY NEWTST"
U 133E, 0000,003D,0180,0800,0000,105F  ;3413 DUM: NEWTST
            ;3414
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0057
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09     Page  10
;                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1925: 1892 1899= 1915= 1916= 1917= 1893 2081=
U 1008  2732= 2733= 2735= 1894 2068= 2069= 1895 2082=
U 1010  2773= 2774= 2777= 1896 2073= 2074= 1897 2550=
U 1018  2092= 2093= 2125= 2126= 2127= 2128= 1898 2551=
U 1020  2142= 2143= 1918 2560= 2144= 2145= 1919 2561=
U 1028  3160= 3161= 3162= 3163= 3165= 1920 2146= 2147=
U 1030  2157= 2158= 2166= 2167= 3168= 3169= 3170= 3171=
U 1038  3182= 3183= 3184= 3185= 3186= 1921 2186= 2187=
U 1040  2188= 2189= 2210= 2211= 3189= 3190= 3191= 3192=
U 1048  2212= 2213= 2222= 2223= 2245= 2246= 1958: 1922
U 1050  3256= 3257= 3259= 3260= 3262= 1923 2250= 2251=
U 1058  2274= 2275= 1961: 1924 2277= 2278= 1964: 1936
U 1060  2282= 2283= 2324= 2325= 3265= 3266= 3267= 3268=
U 1068  3279= 3280= 3281= 3282= 3283= 1937 2327= 2328=
U 1070  2330= 2331= 2334= 2335= 3286= 3287= 3288= 3289=
U 1078  3361= 3362= 3363= 3364= 3365= 1938 2337= 2338=
U 1080  2340= 2341= 2344= 2345= 3368= 3369= 3370= 3371=
U 1088  3383= 3384= 3385= 3386= 3387= 1939 2348= 2349=
U 1090  2394= 2395= 2399= 2400= 3390= 3391= 3392= 3393=
U 1098  2402= 2403= 2404= 2405= 2409= 2410= 2412= 2413=
U 10A0  2414= 2415= 2419= 2420= 2422= 2423= 2424= 2425=
U 10A8  2429= 2430= 2432= 2433= 2495= 2496= 2516= 2517=
U 10B0  2518= 2519= 2521= 2527= 2535= 2536= 2544= 2545=
U 10B8  2601= 2602= 2626= 2627= 2628= 2629= 2630= 2631=
U 10C0  2632= 2633= 2638= 2644= 2654= 2655= 2662= 2663=
U 10C8  2667= 2668= 2669= 2670= 2671= 2672= 2678= 2684=
U 10D0  2692= 2693= 2701= 2702= 2703= 2704= 2708= 2709=
U 10D8  2713= 2714= 2715= 2716= 2721= 2722= 2728= 2729=
U 10E0  2744= 2745= 2748= 2749= 2753= 2754= 2755= 2756=
U 10E8  2763= 2764= 2769= 2770= 2861= 2862= 2873= 2874=
U 10F0  2875= 2876= 2877= 2878= 2879= 2880= 2887= 2889=
U 10F8  2895= 2896= 2903= 2904= 2905= 2906= 2908= 2909=
U 1100  1879: 1880: 1881: 1882: 1883: 1884: 1885: 1886:
U 1108  1887: 1940 2910= 2911= 1888: 1889: 1890: 1891:
U 1110  2912= 2913= 2915= 2916= 2918= 2919= 2925= 2926=
U 1118  2933= 2934= 2936= 2937= 2939= 2940= 2941= 2942=
U 1120  2943= 2944= 2955= 2956= 2963= 2965= 2971= 2972=
U 1128  2979= 2980= 2981= 2982= 2991= 2992= 3000= 3001=
U 1130  3038= 3039= 3040= 3041= 3042= 3049= 3050= 3051=
U 1138  3055= 3056= 3058= 3059= 3066= 3067= 3069= 3070=
U 1140  3072= 3073= 3074= 3075= 3083= 3084= 3088= 3089=
U 1148  3091= 3092= 3093= 3095= 3128= 3129= 3130= 3131=
U 1150  3132= 3133= 3143= 3145= 1941 1986: 3147= 3148=
U 1158  3158= 3159= 3180= 3181= 3226= 3227= 3228= 3229=
U 1160  3230= 3231= 3240= 3241= 3245= 3246= 3254= 3255=
U 1168  3277= 3278= 3323= 3324= 3325= 3326= 3327= 3328=
U 1170  3337= 3338= 3340= 3341= 3344= 3345= 3359= 3360=
U 1178  3381= 3382= 3399= 3400= 2445: 2446: 2447: 2448:
U 1180  3404= 3405= 3409= 3410= 1942 1943 1944 1945
U 1188  1946 1947 1948 1949 1950 1951 1952 1953
U 1190  1954 1955 1956 1957 1959 1960 1996 1997
U 1198  2005 2006 2007 2011 2038 2039 2040 2041
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0058
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09     Page  10

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2042 2043 2044 2065 2066 2067 2070 2071
U 11A8   2072 2075 2076 2077 2078 2091 2094 2095
U 11B0   2102 2103 2104 2105 2113 2114 2115 2148
U 11B8   2149 2174 2190 2191 2192 2203 2204 2205
U 11C0   2206 2207 2208 2209 2214 2215 2216 2217
U 11C8   2218 2219 2220 2221 2224 2225 2237 2238
U 11D0   2239 2240 2241 2242 2243 2244 2247 2248
U 11D8   2249 2265 2266 2267 2268 2269 2270 2271
U 11E0   2272 2273 2276 2279 2280 2281 2292 2301
U 11E8   2302 2303 2304 2305 2306 2307 2308 2309
U 11F0   2310 2311 2312 2313 2314 2315 2316 2317
U 11F8   2318 2319 2320 2321 2449: 2450: 2451: 2452:
U 1200   2322 2326 2332 2336 2342 2346 2350 2382
U 1208   2013: 2383 2384 2385 2386 2387 2388 2389
U 1210   2390 2391 2392 2393 2396 2397 2025: 2398
U 1218   2401 2406 2407 2408 2015: 2411 2416 2417
U 1220   2418 2421 2426 2427 2428 2431 2444 2497
U 1228   2498 2499 2500 2501 2502 2503 2510 2511
U 1230   2512 2513 2514 2515 2520 2528 2529 2530
U 1238   2531 2532 2533 2534 2537 2538 2539 2540
U 1240   2541 2542 2543 2546 2547 2548 2549 2559
U 1248   2562 2603 2604 2605 2606 2607 2608 2016:
U 1250   2609 2014: 2610 2017: 2611 2612 2613 2614
U 1258   2615 2616 2617 2618 2619 2634 2024: 2635
U 1260   2636 2645 2646 2647 2648 2649 2650 2651
U 1268   2652 2664 2665 2666 2673 2674 2675 2676
U 1270   2685 2686 2687 2688 2689 2690 2705 2706
U 1278   2707 2710 2711 2712 2018: 2019: 2020: 2717
U 1280   2718 2719 2723 2724 2726 2727 2746 2747
U 1288   2750 2751 2752 2757 2758 2759 2760 2021:
U 1290   2022: 2761 2765 2766 2767 2768 2863 2864
U 1298   2865 2866 2881 2882 2883 2884 2885 2886
U 12A0   2890 2891 2892 2893 2894 2907 2914 2920
U 12A8   2921 2922 1987: 2923 2924 2935 2938 2945
U 12B0   2946 2947 2948 2949 2950 2951 2952 2953
U 12B8   2954 2957 2958 2959 2960 2961 2962 2966
U 12C0   2967 2968 2969 2970 2983 2984 2985 2986
U 12C8   2987 2988 2989 2990 2993 2994 2995 2996
U 12D0   2997 2998 2999 3052 3053 3054 3057 3068
U 12D8   3071 3085 3086 2023: 3087 2026: 3090 3134
U 12E0   3135 3136 3137 3138 3139 3140 3141 3142
U 12E8   3146 3149 3150 3151 3152 3153 3154 3155
U 12F0   3156 3157 3167 3172 3173 3174 3175 3176
U 12F8   3177 3178 3179 3188 3193 3232 3233 3234
U 1300   3235 3236 3237 3238 3239 3242 3243 3244
U 1308   3247 3248 3249 3250 3251 3252 3253 3264
U 1310   3269 3270 3271 3272 3273 3274 3275 3276
U 1318   3285 3290 3329 3330 3331 3332 3333 3334
U 1320   3335 3336 3339 3342 3343 3346 3347 3348
U 1328   3349 3350 3351 3352 3353 3354 3355 3356
U 1330   3357 3358 3367 3372 3373 3374 3375 3376
U 1338   3377 3378 3379 3380 3389 3394 3413
U 1340 - 13EF Unused
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0059
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09     Page  10

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 13F0  1970: 1971: 1972: 1973: 1974: 1975: 1976: 1977:
U 13F8  1978: 1979: 1980: 1981: 1982: 1983: 1984: 1985:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0060
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09     Page  10

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR3D        847

Used               847
Remaining

Total microwords used in memory U: 847
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                    SEQ 0061
ESKAR3D.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:09   Page  10


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR3D.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR3E.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 0062

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1855 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1856   MICRO TRAP HANDLER
; 1924   ASCII MESSAGES
; 1950 CONSOLE INTERFACE ROUTINES
; 1983   ERLOOP
; 1987   ERROR1 WITH PARAMETER
; 2008   ERROR1
; 2014   ERROR 2 WITH PARAMETER
; 2034   ERROR2
; 2040   CALLCON (CONSOLE CALL)
; 2046   RESERVED CONTROL STORE
; 2070   SUBTEST NUMBER INCREMENT
; 2081   SET ARGUMENT COUNT
; 2092 DIAGNOSTIC SPECIFIC SUBROUTINES
; 2093   DISABLE CACHE
; 2103   SELECT CONTROLLER
; 2142   64K OR 256K CHIPS
; 2196   GET MEMORY SIZE
; 2255   DISABLE SBI CYCLES
; 2265   CLEAR SBI.ERR REG
; 2274   ENABLE SBI FAULT INTERRUPTS
; 2291   CHECK FOR INTERRUPT
; 2324   CHECK FOR TIME-OUT INTERRUPT
; 2348   SBI UNJAM
; 2372   GENERATE I/O ADDRESS BASED ON TR
; 2413   SILO EXPECTED WRITE DATA
; 2437   SILO EXPECTED COMMAND ADDRESS DATA
; 2466   SILO WRITE REF COMMAND ADDRSS DATA
; 2485   SILO UNLOCK AND WAIT
; 2501   SILO DATA TO RC REGS
; 2533   RC (SILO) DATA CHECK
; 2596   TYPE DEVICE NAME
; 2808   TYPE DEV NAME AND TR MESSAGE
; 2831   PRINT ASCII
; 2868   DECREMENT TYPE-OUT FLAG
; 2879   SBIDAT (FOR NEXUS RESPONSE TEST
; 2895   FIND CPU ID FROM SILO
; 2922   SILO LOCK PATTERN FOR RD DATA
; 2939 TEST 17B SBI.HOLD TEST
; 3019 TEST 17C TIMEOUT TEST
; 3120 TEST 17D CPU ID TEST
; 3208 TEST 17E COMMAND ADDRESS TEST
; 3438 TEST 17F SBI NEXUS RESPONSE TEST
; 3611 TEST 180 AT LEAST ONE MEMORY CONTROLLER PRESENT TEST
; 3701 TEST 181 TIMEOUT COUNT TEST
; 3800 TEST 182 RESERVED
; 3805 TEST 183 RESERVED
; 3810 TEST 184 RESERVED
; 3815 DUMMY NEWTST

```
          ;1  .NOLIST
          ;1804  .LIST
          ;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR3E
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; MODIFIED BY: PAUL HAKALA
;1836 ;
;1837 ; 1.0 December 6,1982  Paul Hakala
;1838 ;    Modified message NUMBER 10 to
;1839 ;    print properly took out A "?" which
;1840 ;     is a illegal character for Radix 50
;1841 ;
;1842 ; 1.1 March 10, 1983  Paul Hakala
;1843 ;   Modified the GET.MEMORY.SIZE routine to calculate
;1844 ;   the correct memory address when 256k ram chips are
;1845 ;   present.
;1846 ;
;1847 ;
;1848 ;
;1849 ;
;1850 ;
;1851 ;*****************************************************************************
;1852 ;
```

```
            ;1853  .PAGE
            ;1854
            ;1855  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1856  .TOC "    MICRO TRAP HANDLER"
            ;1857 ;*
            ;1858 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1859 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1860 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1861 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1862 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1863 ;
            ;1864 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1865 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1866 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1867 ; R[0F] AND DO A RETURN0.
            ;1868 ;
            ;1869 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1870 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1871 ; TO THE CONSOLE.
            ;1872 ;
            ;1873 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1874 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7001  ;1875 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7001  ;1876 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7001  ;1877 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7001  ;1878 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7001  ;1879 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7001  ;1880 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7001  ;1881 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7001  ;1882 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7001  ;1883 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D, 0000.003C.8980.0800.0084.7001  ;1884 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7001  ;1885 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1886 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.102B  ;1887 TRPH0: Q_ID[USTACK]
U 102B, 0C00.003C.01E0.0800.0000.1033  ;1888  Q_D,D_Q
U 1033, 0000.003C.8180.3C00.0000.1036  ;1889  ID[USTACK]_D
U 1036, 0C00.003C.01E0.0800.0000.103E  ;1890  Q_D,D_Q
U 103E, 0000.003C.01C0.0A78.0000.1046  ;1891  Q_R[0F]
U 1046, 0001.2024.0180.0800.0010.105B  ;1892  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 105B, 0000.013C.0180.0800.0000.1002  ;1893  Z?     ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1894 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1895 ;*
            ;1896 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1897 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1898 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1899 ;
            ;1900 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1901 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1902 ;       FAULT STATUS REGISTER
            ;1903 ;       SBI ERROR REGISTER
            ;1904 ;       TIMEOUT ADDRESS REGISTER
            ;1905 ;       MAINTENANCE REGISTER
            ;1906 ;       CACHE PARITY REGISTER
            ;1907 ;       TB ERROR REGISTER 1
```

```
            :1908 ;       TB ERROR REGISTER 0
            :1909 ;-
U 1003, 0018.0038.1D80.09E8.0000.102C  ;1910 TRPH1: RC[0D]_SC
U 102C, 0000.003D.D980.0800.0084.722F  ;1911 =0 SETRCF[.9]
U 102D, 0000.003C.49F0.2C00.0000.105F  ;1912  Q_ID[TBER0]
U 105F, 0001.203C.4DF0.2DB0.0000.1109  ;1913  RC[6]_Q,Q_ID[TBER1]
U 1109, 0001.203C.65F0.2DB8.0000.1154  ;1914  RC[7]_Q,Q_ID[SBI.ERR]
U 1154, 0001.203C.6DF0.2DD8.0000.1212  ;1915  RC[0B]_Q,Q_ID[FAULT]
U 1212, 0001.203C.75F0.2DE0.0000.1213  ;1916  RC[0C]_Q,Q_ID[MAINT]
U 1213, 0001.203C.79F0.2DC8.0000.1214  ;1917  RC[9]_Q,Q_ID[PARITY]
U 1214, 0001.203C.69F0.2DC0.0000.1215  ;1918  RC[8]_Q,Q_ID[TIME.ADDR]
U 1215, 0001.203C.81F0.2DD0.0000.1000  ;1919  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1229  ;1920 1000: RC[0E]_Q,ERROR1
            :1921
            :1922
```

```
;1923  .PAGE
;1924  .TOC "   ASCII MESSAGES"
;1925 ;*
;1926 ; THE FOLLOWING ARE THE MESSAGES USED BY THIS OVERLAY.
;1927 ;
;1928 ;& 01./MS780-A  4K CHIP/
;1929 ;& 02./MS780-C 16K CHIP/
;1930 ;& 03./UNKNOWN DEVICE/
;1931 ;& 04./RH780/
;1932 ;& 05./DW780/
;1933 ;& 06./DR780/
;1934 ;& 07./CI780/
;1935 ;& 08./MA780/
;1936 ;& 09./MS780E ERR/
;1937 ;& 0A./MS780E 64K CHIP/
;1938 ;& 0B./MS780H 256K CHIP/
;1939 ;& 0C./ AT TR /
;1940 ;& 0D./  MAX ADDRESS+1= /
;1941 ;& 0E./  PORT NUMBER= /
;1942 ;& 0F./CPU TR = /
;1943 ;& 10./NO MEMORY CNTRLLRS/
;1944 ;& 11./DEP MS780EH ADDR IN RC0, TYPE LO/
;1945 ;& 12./  LOWER CNTRLLR/
;1946 ;& 13./  UPPER CNTRLLR/
;1947 ;& 14./ MISCONFIGURED/
;1948
```

```
            ;1949   .PAGE
            ;1950   .TOC " CONSOLE INTERFACE ROUTINES"
            ;1951 ;*
            ;1952 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1953 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1954 ;  NEWTST
            ;1955 ;  ERLOOP
            ;1956 ;  ERROR1
            ;1957 ;  ERROR2
            ;1958 ;
U 1216. 0000.003C.65F8.0800.0084.7217   ;1959 SCOPE: SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 1217. 0818.0038.8D80.0800.0000.1218   ;1960  D_K[.1F]    ; ...
U 1218. 0D00.003C.0180.0800.0000.1219   ;1961  D_DAL.SC
U 1219. 0000.003C.3D80.3C00.0000.121A   ;1962  ID[PSL]_D    ; WRITE THE PSL
U 121A. 0818.0038.0580.0800.0000.121B   ;1963  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 121B. 0D00.003C.0180.0800.0000.121C   ;1964  D_DAL.SC
U 121C. 0F00.003C.3180.3C00.0000.121D   ;1965  ID[CES]_D,D_0    ; CLEAR THE CES REGISTER
U 121D. 0000.003C.5D80.3C00.0000.121E   ;1966  ID[ACC.CS]_D    ; SHUT OFF THE FLOATING POINT
U 121E. 0000.003C.3980.3C00.0000.121F   ;1967  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 121F. 0000.003C.2980.3C00.0000.1220   ;1968  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1220. 0000.003C.4980.3C00.0000.1034   ;1969  ID[TBER0]_D    ; SHUT OFF THE TB
U 1034. 0000.003D.0180.0800.0000.1266   ;1970 =0 CALL.J/HLD.UNJ
U 1035. 0000.003C.0180.0800.4000.1221   ;1971  IEK/ISTR
U 1221. 0818.0038.C180.0800.0000.1222   ;1972  D_K[.FFFF]
U 1222. 0801.0028.6580.3C00.0000.1223   ;1973  ID[SBI.ERR]_D,D_NOT.D
U 1223. 0F00.003C.6D80.3C00.0000.1224   ;1974  ID[FAULT]_D,D_0
U 1224. 0000.003C.6D80.3C00.0000.1225   ;1975  ID[FAULT]_D
U 1225. 0000.003C.6580.3C00.0000.1226   ;1976  ID[SBI.ERR]_D
U 1226. 0003.003C.0180.09F0.0000.1227   ;1977  RC[0E]_0
U 1227. 0003.003C.0180.0AF8.0000.103C   ;1978  R[0F]_0
U 103C. 0000.003D.0180.0800.0000.1232   ;1979 =0 CALL.J/COFF    ; TURN OFF THE CACHE
U 103D. 0F03.003C.0180.09E8.0000.105E   ;1980  RC[0D]_0,D_0,J/CALLCON    ; CALL THE CONSOLE
            ;1981
```

```
            ;1982  .PAGE
            ;1983  .TOC "    ERLOOP"
U 1228. 0810,0038,0980,0800,0000,105E  ;1984 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
            ;1985
```

```
            ;1986  .PAGE
            ;1987  .TOC "   ERROR1 WITH PARAMETER"
            ;1988 ;**
            ;1989 ; FUNCTIONAL DESCRIPTION:
            ;1990 ;
            ;1991 ; This subroutine takes  as parameter the number of arguments
            ;1992 ; to be printed to the console in the case of an error logout.
            ;1993 ;
            ;1994 ; CALLING CONSTRAINT:
            ;1995 ;
            ;1996 ; =0
            ;1997 ;INPUTS:
            ;1998 ;
            ;1999 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2000 ;--
            ;2001 =0
            ;2002 ERR1.ARG:
U 1044, 0000,003D,0180,0800,0000,122F  ;2003  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1045, 0000,003C,0180,0800,0000,1229  ;2004  J/ERROR1  ; Go to ERROR1
            ;2005 ;
            ;2006 ;
```

```
            ;2007  .PAGE
            ;2008  .TOC "    ERROR1"
U 1229, 0810,0038,0180,0978,0000,122A  ;2009 ERROR1: D_RC[0F]
U 122A, 0819,0034,4D80,0800,0000,104E  ;2010  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2011 104E: D_D.OR.K[.4],J/CALLCON
            ;2012
```

```
            ;2013  .PAGE
            ;2014  .TOC "   ERROR 2 WITH PARAMETER"
            ;2015 ;++
            ;2016 ; FUNCTIONAL DESCRIPTION:
            ;2017 ;
            ;2018 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2019 ; except that in this case after setting the number of arguments
            ;2020 ; too the console, control is transferred to routine ERROR2.
            ;2021 ;
            ;2022 ; INPUTS:
            ;2023 ;
            ;2024 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2025 ;
            ;2026 ;--
            ;2027 =0
            ;2028 ERR2.ARG:
U 104C, 0000,003D,0180,0800,0000,122F  ;2029  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 104D, 0000,003C,0180,0800,0000,122B  ;2030  J/ERROR2  ; Go to ERROR2
            ;2031 ;
            ;2032 ;
```

```
            ;2033  .PAGE
            ;2034  .TOC "    ERROR2"
U 122B, 0810,0038,0180,0978,0000,122C  ;2035 ERROR2: D_RC[0F]
U 122C, 0819,0034,4D80,0800,0000,105A  ;2036  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2037 105A: D_D.OR.K[.6],J/CALLCON
            ;2038
```

```
            ;2039  .PAGE
            ;2040  .TOC "   CALLCON (CONSOLE CALL)"
            ;2041 105E:
            ;2042 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2043  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2044
```

```
            ;2045  .PAGE
            ;2046  .TOC "   RESERVED CONTROL STORE"
            ;2047 ;*
            ;2048 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2049 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1   ;2050 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2   ;2051 13F1: NOP
U 13F2. 0000.003C.0180.0800.0000.13F3   ;2052 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4   ;2053 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5   ;2054 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6   ;2055 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7   ;2056 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8   ;2057 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9   ;2058 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA   ;2059 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB   ;2060 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC   ;2061 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD   ;2062 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE   ;2063 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF   ;2064 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1155   ;2065 13FF: NOP
U 1155. 0000.003C.0180.0800.0000.12AA   ;2066 1155: NOP
U 12AA. 0000.003C.0180.0800.0000.122D   ;2067 12AA: NOP
            ;2068
```

```
            ;2069   .PAGE
            ;2070   .TOC "   SUBTEST NUMBER INCREMENT"
            ;2071 ;*
            ;2072 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2073 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2074 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2075 ; TYPING IT.
            ;2076 ;-
U 122D. 0810,0038,4D80,0978,0000,122E   ;2077 SUBTST: D_RC[0F],KMX/.FF00
U 122E. 0019,4016,4D80,09F8,0000,0001   ;2078  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2079
```

```
            ;2080  .PAGE
            ;2081  .TOC "   SET ARGUMENT COUNT"
            ;2082 ;+
            ;2083 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2084 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2085 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2086 ;-
U 122F, 0010,0038,01C0,0978,0000,1230  ;2087 SETRCF: Q_RC[0F]
U 1230, 0019,2034,4DC0,0800,0000,1231  ;2088  Q_Q.AND.K[.FF00]
U 1231, 0019,2032,1D80,09F8,0000,0001  ;2089  RC[0F]_Q.OR.SC,RETURN1
            ;2090
```

```
            ;2091  .PAGE
            ;2092  .TOC " DIAGNOSTIC SPECIFIC SUBROUTINES"
            ;2093  .TOC "   DISABLE CACHE"
            ;2094 ;++
            ;2095 ; THIS SUBROUTINE IS CALLED TO SET THE FORCE CACHE MISS
            ;2096 ; BITS FOR EACH GROUP IN THE MAINTENANCE REGISTER ON THE ID BUS.
            ;2097 ;--
U 1232. 0818.0038.0DF8.0800.0000.1233  ;2098 COFF: D_K[.3],Q_0
U 1233. 0000.003C.6180.0800.0084.7234  ;2099  SC_K[.F]
U 1234. 0D00.003C.0180.0800.0000.1235  ;2100  D_DAL.SC
U 1235. 0000.003E.7580.3C00.0000.0001  ;2101  ID[MAINT]_D,RETURN1
```

```
            ;2102  .PAGE
            ;2103  .TOC "   SELECT CONTROLLER"
            ;2104 ;**
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; This routine is used to put the memory system into the
            ;2108 ; specified configuration with respect to controller select.
            ;2109 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2110 ; a RETURN2 is made.
            ;2111 ;
            ;2112 ; CALLING CONSTRAINT:
            ;2113 ;
            ;2114 ; =00
            ;2115 ;
            ;2116 ; INPUTS:
            ;2117 ;
            ;2118 ; d - Contains value to write to bits<2:0> of Register A
            ;2119 ; rc[0e] - Address of Register A
            ;2120 ;
            ;2121 ; SIDE EFFECTS:
            ;2122 ;
            ;2123 ; sc,d,va are destroyed
            ;2124 ;--
            ;2125 SELECT.CNTRLR:
U 1236, 0010,0038,0180,0970,0284,7237  ;2126  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1237, 0801,001C,0180,0800,0000,1238  ;2127  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1238, 0000,003C,0180,A800,0000,1239  ;2128  cache.p_d[long]  ; Write the configuration Register
U 1239, 0000,003C,0180,C800,0000,123A  ;2129  d[long]_cache.p  ; Read CNFG A Reg and
U 123A, 0818,0038,5D80,0800,0000,123B  ;2130  d_k[.7]   ; Get Misconfiguration bits
U 123B, 0000,003C,61F8,0800,0084,723C  ;2131  sc_k[.F],q_0  ; ...
U 123C, 0D00,003C,0180,0800,0000,123D  ;2132  d_dal.sc  ; ... D = x38000
U 123D, 0000,003C,01E0,0800,0000,123E  ;2133  q_d   ; Save mask
U 123E, 0000,003C,0180,C800,0000,123F  ;2134  d[long]_cache.p  ; Read CNFG A Reg and
            ;2135  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 123F, 001D,2034,0180,0800,0010,1240  ;2136   clk.ubcc ; ...
U 1240, 0000,013C,0180,0800,0000,1058  ;2137  z?   ; Branch if no
U 1058, 0000,003E,0180,0800,0000,0001  ;2138 =0 RETURN1   ; Bad configuration
U 1059, 0000,003E,0180,0800,0000,0002  ;2139  RETURN2   ; Configuration ok
            ;2140
```

```
            ;2141   .PAGE
            ;2142   .TOC "   64K OR 256K CHIPS"
            ;2143 ;*****************************************************************
            ;2144 ;++
            ;2145 ;
            ;2146 ; Functional Description:
            ;2147 ; ----------------------
            ;2148 ; This subroutine is used to find the type of chips
            ;2149 ; on the MS780-E/H arrays.  The RETURN modes are as
            ;2150 ; follows:
            ;2151 ;
            ;2152 ; RETURN1 = Error. Mixed 64K and 256K arrays
            ;2153 ;
            ;2154 ; RETURN2 = 64K chips on the array
            ;2155 ;
            ;2156 ; RETURN3 = 256K chips on the array
            ;2157 ;
            ;2158 ; Calling Constraint: =00
            ;2159 ; ------------------
            ;2160 ;
            ;2161 ;
            ;2162 ; Inputs:
            ;2163 ; ------
            ;2164 ;
            ;2165 ; RC[0E] must hold the I/O base address of the MS780-E/H
            ;2166 ; currently under test
            ;2167 ;
            ;2168 ; Outputs:
            ;2169 ; -------
            ;2170 ;
            ;2171 ; NO specific outputs. (See RETURN modes above)
            ;2172 ;
            ;2173 ;
            ;2174 ; Side Effects:
            ;2175 ; ------------
            ;2176 ;
            ;2177 ; The contents of the following registers will be lost:
            ;2178 ;
            ;2179 ; D
            ;2180 ;
            ;2181 ;--
            ;2182 ;*****************************************************************
            ;2183 64K.OR.256K:
U 1241,  0010,003B,0180,0970,0200,1242  ;2184  VA_RC[0E]    ; Point to CNFG Register A
U 1242,  0000,003C,0180,C800,0000,1243  ;2185  D[LONG]_CACHE.P    ; Read the register
U 1243,  0200,003C,0180,0800,0000,1244  ;2186  D_D.RIGHT2    ; Shift received contents two bits
            ;2187        ; ...to the right
U 1244,  0600,003C,0180,0800,0000,1245  ;2188  D_D.RIGHT    ; Shift one more time
U 1245,  0000,193C,0180,0800,0000,100C  ;2189  D1-0?     ; Branch on the RAM Type
U 100C,  0000,003E,0180,0800,0000,0001  ;2190 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D,  0000,003E,0180,0800,0000,0002  ;2191  RETURN2     ; 64K RAMs found
U 100E,  0000,003E,0180,0800,0000,0003  ;2192  RETURN3     ; 256K RAMs found
U 100F,  0000,003E,0180,0800,0000,0001  ;2193  RETURN1     ; Error: missing arrays
            ;2194
```

```
                ;2195  .PAGE
                ;2196  .TOC "   GET MEMORY SIZE"
                ;2197 ;*****************************************************************
                ;2198 ;**
                ;2199 ;
                ;2200 ; Functional Description:
                ;2201 ; ----------------------
                ;2202 ;
                ;2203 ; This subroutine can be used to get the size of the memory.
                ;2204 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
                ;2205 ; increments.
                ;2206 ;
                ;2207 ; Calling Constraint: =0
                ;2208 ; ------------------
                ;2209 ;
                ;2210 ;
                ;2211 ; Inputs:
                ;2212 ; ------
                ;2213 ;
                ;2214 ; RC[0E] Must have the Address of CNFG Register A
                ;2215 ;
                ;2216 ; Outputs:
                ;2217 ; -------
                ;2218 ;
                ;2219 ; Register D will contain upon exit the size of the
                ;2220 ; memory aligned on Mega-byte boundary.
                ;2221 ;
                ;2222 ; Side Effects:
                ;2223 ; ------------
                ;2224 ;
                ;2225 ; The contents of the following registers will be lost:
                ;2226 ;
                ;2227 ; D, SC, Q
                ;2228 ;
                ;2229 ;--
                ;2230 ;*****************************************************************
                ;2231 GET.MEMORY.SIZE:
U 1246, 0010,0038,0180,0970,0200,1247  ;2232  VA_RC[0E]    ; Point to CNFG Reg A
U 1247, 0000,003C,0180,C800,0000,1248  ;2233  D[LONG]_CACHE.P    ; Read the register
U 1248, 001B,0000,D980,0800,0082,1249  ;2234  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 1249, 0D00,003C,0180,0800,0000,124A  ;2235  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 124A, 0819,0034,5580,0800,0000,1004  ;2236  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
                ;2237 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1004, 0000,003D,01E0,0800,0000,1241  ;2238   Q_D    ; Save Memory size bits in Q
U 1005, 0000,003D,0D80,0800,0084,7044  ;2239  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
                ;2240       ; ...arrays on the controller
                ;2241 ;
                ;2242 ; The duplication of the next two return status is necessary so the returning
                ;2243 ; subroutine (64k.or.256k) can be used in other modules without any
                ;2244 ; modifications.
                ;2245 ;
U 1006, 0819,2014,0580,0800,0000,1007  ;2246  D_Q+K[.1]    ; RET2 Increment memory size by one
                ;2247       ; ...(found 1 Meg arrays)
U 1007, 0819,2014,0580,0800,0000,124B  ;2248  D_Q+K[.1]    ; RET3 Increment memory size by 1
                ;2249       ; ...(found 4 Meg arrays)
```

```
U 124B, 0000,003C,21F8,0800,0084,724C  ;2250  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
            ;2251  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 124C, 0D00,003E,0180,0800,0000,0001  ;2252   RETURN1    ; Return with memory size in Register D
            ;2253
```

```
            :2254  .PAGE
            :2255  .TOC "   DISABLE SBI CYCLES"
            :2256 ;++
            :2257 ; THIS SUBROUTINE IS CALLED TO DISABLE SBI CYCLES BY SETTING THE
            :2258 ; APPROPRIATE BIT IN THE SBI MAINTENANCE REGISTER.
            :2259 ;--
U 124D. 0000.003C.8580.0800.0084.724E  ;2260 SOFF: SC_K[.C]
U 124E. 0803.001C.0180.0800.0000.124F  ;2261  ALU_NOT.MASK,D_ALU
U 124F. 0000.003E.7580.3C00.0000.0001  ;2262  ID[MAINT]_D,RETURN1
            :2263
```

```
            ;2264  .PAGE
            ;2265  .TOC "   CLEAR SBI.ERR REG"
            ;2266 ;*************************************************************
            ;2267 ;* THIS ROUTINE CLEARS THE SBI.ERR REGISTER
            ;2268 ;-
U 1250, 081B.0000.0580.0800.0000.1251  ;2269 CLCPTO: D_0-K[.1]
U 1251, 0801.0028.6580.3C00.0000.1252  ;2270  ID[SBI.ERR]_D,D_NOT.D
U 1252, 0000.003E.6580.3C00.0000.0001  ;2271  ID[SBI.ERR]_D,RETURN1
            ;2272
```

```
            ;2273  .PAGE
            ;2274  .TOC "   ENABLE SBI FAULT INTERRUPTS"
            ;2275 ;*****************************************************************
            ;2276 ;*
            ;2277 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2278 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2279 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2280 ;
            ;2281 ; D AND SC GET CLOBBERED
            ;2282 ;-
            ;2283 ;*****************************************************************
U 1253. 0F00,003C,2180,0800,0084,7254  ;2284 ENFLTI: SC_K[.14],D_0
U 1254. 0000,003C,0580,0800,0084,B255  ;2285  SC_SC-K[.1]
U 1255. 0801,001C,0180,0800,0000,1256  ;2286  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 1256. 0600,003C,6D80,3C00,0000,1257  ;2287  ID[FAULT]_D,D_D.RIGHT
U 1257. 0000,003E,6D80,3C00,0000,0001  ;2288  ID[FAULT]_D,RETURN1
            ;2289
```

```
            ;2290  .PAGE
            ;2291  .TOC "    CHECK FOR INTERRUPT"
            ;2292 ;*****************************************************************
            ;2293 ;+
            ;2294 ;       THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
            ;2295 ;       FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
            ;2296 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
            ;2297 ; IF ANY INTERRUPTS OCCURRED:
            ;2298 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
            ;2299 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
            ;2300 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
            ;2301 ;  RETURN1 IS USED TO GET BACK
            ;2302 ; ELSE
            ;2303 ;  RETURN2 IS USED.
            ;2304 ;
            ;2305 ; THE D AND Q REGISTERS GET CLOBBERED.
            ;2306 ;
            ;2307 ;-
            ;2308 ;*****************************************************************************
            ;2309 ;/////////////////////////////////////////////////////////////////////////////
U 1258. 0F00,003C,0180,0800,0000,1259   ;2310 INTER: D_0
U 1259. 0000,003C,3D80,3C00,0000,125A   ;2311  ID[PSL]_D  ; SET LOWEST IPL ACTIVE
U 125A. 0000,003C,0180,0800,4000,125B   ;2312  IEK/ISTR  ; STROBE THE VECTOR
U 125B. 0000,003C,0180,0800,0000,125C   ;2313  NOP    ; WAIT
U 125C. 0000,003C,35F0,2C00,0000,125D   ;2314  Q_ID[VECTOR]
U 125D. 0819,2034,8180,0800,0010,125E   ;2315  ALU_Q.AND.K[.3FF],CLK.UBCC,D_ALU
U 125E. 0000,013C,0180,0800,0000,105C   ;2316  Z?    ; ANY INTERRUPTS?
U 105C. 0000,003C,0180,0800,0000,125F   ;2317 =0 J/ERRDAT   ; AN INTERRUPT OCCURRED
U 105D. 0000,003E,0180,0800,0000,0002   ;2318  RETURN2    ; NO INTERRUPTS
U 125F. 0001,203C,65F0,2DC0,0000,1260   ;2319 ERRDAT: RC[8]_Q,Q_ID[SBI.ERR]
U 1260. 0001,203C,6DF0,2DB8,0000,1261   ;2320  RC[7]_Q,Q_ID[FAULT]
U 1261. 0001,203E,0180,09B0,0000,0001   ;2321  RC[6]_Q,RETURN1
            ;2322
```

```
            ;2323  .PAGE
            ;2324  .TOC "   CHECK FOR TIME-OUT INTERRUPT"
            ;2325 ;//////////////////////////////////
            ;2326 ;       THIS ROUTINE CHECKS FOR AN EXPECTED TIME OUT INTERRUPT.
            ;2327 ; IT IS SIMILAR TO THE ABOVE ROUTINE EXCEPT FOR:
            ;2328 ;  WAITING 528 CYCLES BEFORE STROBING THE INTERRUPT
            ;2329 ;  RETURN1 IF NO INTERRUPT OCCURRED
            ;2330 ;  RETURN2 IF THE WRONG INTERRUPT VECTOR IS GENERATED
            ;2331 ;  RETURN3 IF A CPU TIME OUT INTERRUPT OCCURRED
U 1262. 0018.0038.95C0.0800.0000.1263  ;2332 WINTER: Q_K[.B0]
U 1263. 0001.203C.0180.0800.0010.1264  ;2333 WNTERA: ALU_Q,CLK.UBCC  ; WAIT 528. CYCLES
U 1264. 0000.013C.0180.0800.0000.1084  ;2334  Z?    ; ...
U 1084. 0019.2000.05C0.0800.0000.1263  ;2335 =0 Q_Q-K[.1],J/WNTERA ; ...
U 1085. 0000.003C.0180.0800.0000.1008  ;2336  NOP
U 1008. 0000.003D.0180.0800.0000.1258  ;2337 =00 CALL,J/INTER
U 1009. 0000.003C.0180.0800.0000.100B  ;2338  J/CHKVCT  ; RETURN1(ABOVE) MEANT AN INTERRUPT
            ;2339      ; OCCURRED(D HAS MASKED VECTOR REGISTER)
U 100A. 0000.003C.0180.0800.0000.125F  ;2340  J/ERRDAT  ; RETURN2(ABOVE)MEANT NO INTERRUPT
            ;2341      ; GO BACK,GET DATA AND LET IT RETURN1
U 100B. 0019.0000.A580.0800.0010.1265  ;2342 CHKVCT: ALU_D-K[.60],CLK.UBCC ; INTERRUPT VECTOR FOR TIME OUT=60
U 1265. 0000.013C.0180.0800.0000.1086  ;2343  Z?
U 1086. 0000.003E.0180.0800.0000.0002  ;2344 =0 RETURN2   ; WRONG VECTOR
U 1087. 0000.003E.0180.0800.0000.0003  ;2345  RETURN3   ; CORRECT VECTOR WAS GENERATED
            ;2346
```

```
            ;2347  .PAGE
            ;2348  .TOC "   SBI UNJAM"
            ;2349 ;*****************************************************************
            ;2350 ;*
            ;2351 ;       THIS ROUTINE PERFORMS AN UNJAM SEQUENCE ON THE SBI.
            ;2352 ;-
            ;2353 ;*****************************************************************
            ;2354 HLD.UNJ:
U 1266. 0818.0038.1180.8000.0000.1267  ;2355  MCT/SBI.HOLD,D_K[.4] ; ASSERT HOLD FOR 16 CYCLES
U 1267. 0001.003C.0180.8000.0010.1268  ;2356 HLD0: MCT/SBI.HOLD,ALU_D,CLK.UBCC
U 1268. 0819.0100.0580.8000.0000.1088  ;2357  MCT/SBI.HOLD,Z?,D_D-K[.1]
U 1088. 0000.003C.0180.8000.0000.1267  ;2358 =0 MCT/SBI.HOLD,J/HLD0
U 1089. 0000.003C.0180.8000.0000.1269  ;2359  MCT/SBI.HOLD
U 1269. 0818.0038.1180.8800.0000.126A  ;2360 UNJAM1: MCT/SBI.HOLD+UNJAM,D_K[.4] ; ASSERT HOLD+UNJAM FOR 16 CYCLES
U 126A. 0001.003C.0180.8800.0010.126B  ;2361 HLD1: MCT/SBI.HOLD+UNJAM,ALU_D,CLK.UBCC
U 126B. 0819.0100.0580.8800.0000.108A  ;2362  MCT/SBI.HOLD+UNJAM,Z?,D_D-K[.1]
U 108A. 0000.003C.0180.8800.0000.126A  ;2363 =0 MCT/SBI.HOLD+UNJAM,J/HLD1
U 108B. 0000.003C.0180.8800.0000.126C  ;2364  MCT/SBI.HOLD+UNJAM
U 126C. 0818.0038.1180.8000.0000.126D  ;2365 UNJAM2: MCT/SBI.HOLD,D_K[.4] ; ASSERT HOLD FOR 16 CYCLES
U 126D. 0001.003C.0180.8000.0010.126E  ;2366 HLD2: MCT/SBI.HOLD,ALU_D,CLK.UBCC
U 126E. 0819.0100.0580.8000.0000.108C  ;2367  MCT/SBI.HOLD,Z?,D_D-K[.1]
U 108C. 0000.003C.0180.8000.0000.126D  ;2368 =0 MCT/SBI.HOLD,J/HLD2
U 108D. 0000.003E.0180.8000.4000.0001  ;2369  IEK/ISTR,MCT/SBI.HOLD,RETURN1
            ;2370
```

```
            ;2371   .PAGE
            ;2372   .TOC "    GENERATE I/O ADDRESS BASED ON TR"
            ;2373 ;*****************************************************************
            ;2374 ;*
            ;2375 ;       THIS SUBROUTINE CALCULATES I/O ADDRESSES CORRESPONDING
            ;2376 ; TO DIFFERENT TR LEVELS ON THE SBI.
            ;2377 ;   I.E. TR 1 => 20002000
            ;2378 ;     TR 2 => 20004000
            ;2379 ;     TR 3 => 20006000
            ;2380 ;      .
            ;2381 ;      .
            ;2382 ;      .
            ;2383 ;     TR F => 2001E000
            ;2384 ; THE ADDRESS IS RETURNED IN THE D REGISTER
            ;2385 ; THE TR LEVEL IS RETURNED IN THE Q-REGISTER
            ;2386 ; SC IS CLOBBERED
            ;2387 ;-
            ;2388 ;*****************************************************************
            ;2389 =0000
U 1010,  0F00,003C,0180,0800,0000,1270   ;2390 TR0: D_0,J/TRXX
U 1011,  0818,0038,0580,0800,0000,1270   ;2391 TR1: D_K[.1],J/TRXX
U 1012,  0818,0038,0980,0800,0000,1270   ;2392 TR2: D_K[.2],J/TRXX
U 1013,  0818,0038,0D80,0800,0000,1270   ;2393 TR3: D_K[.3],J/TRXX
U 1014,  0818,0038,1180,0800,0000,1270   ;2394 TR4: D_K[.4],J/TRXX
U 1015,  0818,0038,1180,0800,0000,126F   ;2395 TR5: D_K[.4],J/ADD1
U 1016,  0818,0038,D580,0800,0000,1270   ;2396 TR6: D_K[.6],J/TRXX
U 1017,  0818,0038,5D80,0800,0000,1270   ;2397 TR7: D_K[.7],J/TRXX
U 1018,  0818,0038,0180,0800,0000,1270   ;2398 TR8: D_K[.8],J/TRXX
U 1019,  0818,0038,D980,0800,0000,1270   ;2399 TR9: D_K[.9],J/TRXX
U 101A,  0818,0038,F580,0800,0000,1270   ;2400 TRA: D_K[.A],J/TRXX
U 101B,  0818,0038,F580,0800,0000,126F   ;2401 TRB: D_K[.A],J/ADD1
U 101C,  0818,0038,8580,0800,0000,1270   ;2402 TRC: D_K[.C],J/TRXX
U 101D,  0818,0038,8980,0800,0000,1270   ;2403 TRD: D_K[.D],J/TRXX
U 101E,  0818,0038,8980,0800,0000,126F   ;2404 TRE: D_K[.D],J/ADD1
U 101F,  0818,0038,6180,0800,0000,1270   ;2405 TRF: D_K[.F],J/TRXX
U 126F,  0819,0014,0580,0800,0000,1270   ;2406 ADD1: D_D+K[.1]
U 1270,  0000,003C,89E0,0800,0084,7271   ;2407 TRXX: SC_K[.D],Q_D
U 1271,  0D00,003C,8D80,0800,0084,7272   ;2408  D_DAL.SC,SC_K[.1F]
U 1272,  0000,003C,0980,0800,0084,B273   ;2409  SC_SC-K[.2]
U 1273,  0801,001E,0180,0800,0000,0001   ;2410  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2411
```

```
            ;2412  .PAGE
            ;2413  .TOC "   SILO EXPECTED WRITE DATA"
            ;2414 ;*****************************************************************
            ;2415 ;*
            ;2416 ;       THIS ROUTINE PLACES THE EXPECTED DATA PATTERN FOR
            ;2417 ; CPU WRITE DATA AS IT APPEARS IN THE SILO.
            ;2418 ;
            ;2419 ; D IS RETURNED WITH 00XX,XXX1,0111,1100,XXXX,XXXX,XXXX,XXXX
            ;2420 ;  X'S ARE DEPENDENT ON CPU'S ID AND TR LEVEL
            ;2421 ; SC,Q,R[0],RC[0E]-RC[0] GET CLOBBERED
            ;2422 ;-
            ;2423 ;*****************************************************************
            ;2424 =0
U 108E. 0000.003C.01F8.0800.0084.708F   ;2425 WDPAT: SC_K[.8],Q_0
U 108F. 0818.0038.9D80.0800.0000.1274   ;2426  D_K[.7C]
U 1274. 0D00.003C.0180.0800.0000.1275   ;2427  D_DAL.SC
U 1275. 0819.0030.0580.0800.0000.1276   ;2428  D_D.OR.K[.1]
U 1276. 0B00.003C.0180.0800.0000.1090   ;2429  D_D.SWAP
U 1090. 0001.003D.0180.0A80.0000.1092   ;2430 =0 CALL,J/CPUCA,R[0]_D
U 1091. 0000.003C.01C0.0A00.0000.1277   ;2431  Q_R[0]
U 1277. 081D.0030.B980.0800.0084.7278   ;2432  D_D.OR.Q,SC_K[.19]
U 1278. 0000.003C.0980.0800.0084.B279   ;2433  SC_SC-K[.2]
U 1279. 0801.0036.0180.0800.0000.0001   ;2434  ALU_D.AND.MASK,D_ALU,RETURN1
            ;2435
```

```
            ;2436  .PAGE
            ;2437  .TOC "   SILO EXPECTED COMMAND ADDRESS DATA"
            ;2438 ;*****************************************************************
            ;2439 ;*
            ;2440 ;       THIS ROUTINE PLACES THE EXPECTED DATA PATTERN FOR
            ;2441 ; CPU COMMAND ADDRESS AS IT APPEARS IN THE SILO.
            ;2442 ; IT GENERATES THE ID AND TAG FIELDS ONLY, I.E. IT IS
            ;2443 ; UP TO THE CALLING ROUTINE TO "OR IN" THE FUNCTION CODE.
            ;2444 ;
            ;2445 ; D IS RETURNED WITH THE VALUE 00XX,XXX0,1100,XXXX,XXXX,XXXX,XXXX
            ;2446 ;  X'S ARE DEPENDENT ON CPU'S ID AND TR LEVEL
            ;2447 ; SC , Q,RC[0E]-RC[0] GET CLOBBERED
            ;2448 ;-
            ;2449 ;*****************************************************************
            ;2450 =0
U 1092. 0000.003C.8980.0800.0084.7093   ;2451 CPUCA: SC_K[.D]
U 1093. 0003.001C.0180.0AF8.0000.1094   ;2452  R[0F]_NOT.MASK
U 1094. 0000.003D.0180.0800.0000.1280   ;2453 =0 CALL,J/SILCLR
U 1095. 0000.003C.0180.0800.0000.1096   ;2454  NOP
U 1096. 0000.003D.0180.0800.0000.1010   ;2455 =0 CALL,J/TR0
U 1097. 0001.003C.7180.3C00.0200.1098   ;2456  ID[COMP]_D,VA_D
U 1098. 0000.003D.0180.C800.0000.10A0   ;2457 =0 CALL,J/SLOLCK,MCT/READ.P
U 1099. 0000.003C.0180.0800.0000.109A   ;2458  NOP
U 109A. 0000.003D.0180.0800.0000.1250   ;2459 =0 CALL,J/CLCPTO
U 109B. 0810.0038.0180.0958.0000.127A   ;2460  D_RC[0B]
U 127A. 0000.003C.2180.0800.0084.727B   ;2461  SC_K[.14]
U 127B. 0000.003C.0980.0800.0084.B27C   ;2462  SC_SC-K[.2]
U 127C. 0801.0036.0180.0800.0000.0001   ;2463  ALU_D.AND.MASK,D_ALU,RETURN1
            ;2464
```

```
            ;2465  .PAGE
            ;2466  .TOC "   SILO WRITE REF COMMAND ADDRSS DATA"
            ;2467 ;*****************************************************************
            ;2468 ;*
            ;2469 ;       THIS ROUTINE GENERATES THE EXPECTED DATA PATTERN FOR
            ;2470 ; A CPU WRITE REFERENCE COMMAND ADDRESS AS IT APPEARS IN THE SILO.
            ;2471 ;
            ;2472 ; D IS RETURNED WITH 00XX,XXX0,1100,1000,XXXX,XXXX,XXXX,XXXX.
            ;2473 ;  X'S ARE DEPENDENT ON THE CPU'S ID AND TR LEVEL
            ;2474 ; SC, Q,RC[0E]-RC[0] GET CLOBBERED
            ;2475 ;-
            ;2476 ;*****************************************************************
            ;2477 =0
U 109C, 0000,003D,0180,0800,0000,1092  ;2478 WRITCA: CALL,J/CPUCA
U 109D, 0000,003C,2180,0800,0084,727D  ;2479  SC_K[.14]
U 127D, 0000,003C,0580,0800,0084,B27E  ;2480  SC_SC-K[.1]
U 127E, 0801,001C,0180,0800,0000,127F  ;2481  D_D.ORNOT.MASK
U 127F, 0819,0032,0580,0800,0000,0001  ;2482  D_D.OR.K[.1],RETURN1
            ;2483
```

```
            ;2484  .PAGE
            ;2485  .TOC "    SILO UNLOCK AND WAIT"
            ;2486 ;*****************************************************************
            ;2487 ;+
            ;2488 ; THIS ROUTINE UNLOCKS THE SILO,AND WAITS FOR 16 CYCLES
            ;2489 ; TO CLEAR IT OUT.
            ;2490 ;-
            ;2491 ;*****************************************************************
U 1280. 0F00.003C.0180.0800.0000.1281   ;2492 SILCLR: D_0
U 1281. 0000.003C.7180.3C00.0000.1282   ;2493  ID[COMP]_D
U 1282. 0818.0038.1180.0800.0000.1283   ;2494  D_K[.4]
U 1283. 0001.003C.0180.0800.0010.1284   ;2495 CLR0: ALU_D,CLK.UBCC
U 1284. 0000.013C.0180.0800.0000.109E   ;2496  Z?
U 109E. 0819.0000.0580.0800.0000.1283   ;2497 =0 J/CLR0,D_D-K[.1]
U 109F. 0000.003E.0180.0800.0000.0001   ;2498  RETURN1
            ;2499
```

```
            ;2500  .PAGE
            ;2501  .TOC "   SILO DATA TO RC REGS"
            ;2502 ;*****************************************************************
            ;2503 ;*
            ;2504 ;       THIS ROUTINE PLACES THE LAST 15 LOCATIONS OF THE SILO
            ;2505 ; IN RC[0E] THRU RC[0].
            ;2506 ;
            ;2507 ;  THE STATE IN WHICH THE CALL IS DONE, AND
            ;2508 ;  THE PRECEDING STATE MUST LOOK EXACTLY AS FOLLOWS:
            ;2509 ;
            ;2510 ;   ID[COMP]_D,...
            ;2511 ;   CALL,J/SLOLCK,MCT/XXXXX
            ;2512 ;
            ;2513 ;   WHERE XXXXX IS ANY REFERENCE.
            ;2514 ;
            ;2515 ;-
            ;2516 ;*****************************************************************
            ;2517 =0
U 10A0, 0818,0038,1180,0800,0000,10A1  ;2518 SLOLCK: D_K[.4]   ; PROCEED TO WAIT A TOTAL OF 16 CYCLES
U 10A1, 0001,003C,0180,0800,0010,1285  ;2519 L1: ALU_D,CLK.UBCC  ; ...
U 1285, 0819,0100,0580,0800,0000,10A2  ;2520  Z?,D_D-K[.1]
U 10A2, 0000,003C,0180,0800,0000,10A1  ;2521 =0 J/L1   ; ...
U 10A3, 0000,003C,6180,0800,0084,7286  ;2522  SC_K[.F]  ; START FOR PLACING SILO IN RC REGISTERS
U 1286, 0000,003C,0580,0800,0084,B287  ;2523 L2: SC_SC-K[.1]  ; STARTING WITH RC[0E]
U 1287, 0000,003C,61F0,2C00,0000,1288  ;2524  Q_ID[SILO]
U 1288, 0001,203C,0180,0838,0000,1289  ;2525  RC(SC)_ALU,ALU_Q
U 1289, 0018,0038,1D80,0800,0010,128A  ;2526  ALU_SC,CLK.UBCC
U 128A, 0000,013C,0180,0800,0000,10A4  ;2527  Z?
U 10A4, 0000,003C,0180,0800,0000,1286  ;2528 =0 J/L2
U 10A5, 0F00,003C,0180,0800,0000,128B  ;2529  D_0
U 128B, 0000,003E,7180,3C00,0000,0001  ;2530  ID[COMP]_D,RETURN1 ; UNLOCK SILO
            ;2531
```

```
                ;2532  .PAGE
                ;2533  .TOC "    RC (SILO) DATA CHECK"
                ;2534 ;*****************************************************************
                ;2535 ;*
                ;2536 ;       THIS ROUTINE CHECKS THE RC SCRATCH PADS FOR CORRECT
                ;2537 ; COMMAND ADDRESS AND CORRECT NUMBER OF RETRY'S AFTER THE
                ;2538 ; SCRATCH PADS HAVE BEEN LOADED WITH SILO DATA USING THE
                ;2539 ; SUBROUTINE NAMED "SLOLCK".
                ;2540 ;
                ;2541 ; WHEN THIS ROUTINE IS CALLED, THE EXPECTED COMMAND/ADDRESS
                ;2542 ; PATTERN (AS IT APPEARS IN THE SILO) MUST BE IN D.
                ;2543 ;
                ;2544 ; IF NO SECONDARY DATA PATTERN IS EXPECTED, CALL THIS ROUTINE
                ;2545 ; WITH STATE REGISTER=0
                ;2546 ; OTHERWISE STATE REGISTER =1
                ;2547 ;
                ;2548 ; IF WRITE OR INTERLOCK REFERENCE, EXPECTED SECONDARY DATA PATTERN
                ;2549 ; (AS IT APPEARS IN THE SILO) MUST BE IN R[0E].
                ;2550 ;
                ;2551 ; THE EXPECTED NUMBER OF CORRECT COMMAND ADDRESSES MUST BE
                ;2552 ; IN R[0D]
                ;2553 ;
                ;2554 ; IF AN ERROR OCCURRS:
                ;2555 ;  RC[0]_ID[SBI.ERR]
                ;2556 ;  RC[1]_ID[FAULT]
                ;2557 ;  RETURN1 IS USED.
                ;2558 ;
                ;2559 ; IF NO ERRORS OCCURR, RETURN2 IS USED.
                ;2560 ;
                ;2561 ; SC AND Q GET CLOBBERRED.
                ;2562 ;-
                ;2563 ;*****************************************************************
U 128C. 0003,003C,61F8,0A80,0084,728D  ;2564 SILRTN: SC_K[.F],Q_0,R[0]_0
U 128D. 0000,003C,0580,0800,0084,B28E  ;2565 SRLP: SC_SC-K[.1]
U 128E. 0010,0038,0180,0830,0010,128F  ;2566  ALU_RC(SC),CLK.UBCC ; NULL OR NR ENTRY?
U 128F. 0000,013C,0180,0800,0000,10A6  ;2567  Z?
U 10A6. 0000,003C,0180,0800,0000,1290  ;2568 =0 J/CAORWD  ; NOT NULL, SEE IF GOOD C/A OR WRITE DATA
U 10A7. 0000,003C,0180,0800,0000,1296  ;2569  J/ENDLP   ; SEE IF DONE
U 1290. 0000,003C,0180,0A00,0010,1291  ;2570 CAORWD: ALU_R[0],CLK.UBCC ; WRITE DATA FLAG SET?
U 1291. 0000,013C,0180,0800,0000,10A8  ;2571  Z?   ; BR IF NO
U 10A8. 0000,003C,0180,0A70,0000,1294  ;2572 =0 J/EQWD,LAB_R[0E] ; CHECK WRITE OR INTLK TAG THIS CYCLE
U 10A9. 0011,0000,0180,0800,0010,1292  ;2573 EQCA: ALU_D-LC,CLK.UBCC ; C/A OK?
U 1292. 0000,013C,0180,0800,0000,10AA  ;2574  Z?   ; BR IF YES
U 10AA. 0000,003C,0180,0800,0000,1299  ;2575 =0 J/CAERR
U 10AB. 0019,2014,05C0,0800,0000,1293  ;2576  Q_Q+K[.1]  ; ADD 1 TO NUM OF GOOD CA'S
U 1293. 0000,173C,0180,0A70,0000,102E  ;2577  STATE0?,LAB_R[0E] ; SECONDARY DATA EXPECTED?,LATCH EXP. DATA
U 102E. 0000,003C,0180,0800,0000,1296  ;2578 =1110 J/ENDLP   ; NO
U 102F. 0018,0038,0580,0A80,0000,1296  ;2579  J/ENDLP,R[0]_K[.1] ; YES SET FLAG FOR NEXT LOOP
U 1294. 0010,0000,0180,0800,0010,1295  ;2580 EQWD: ALU_LA[A-B]LC,CLK.UBCC  ; WRITE OR INTLK DATA OK?
U 1295. 0000,013C,0180,0800,0000,10AC  ;2581  Z?
U 10AC. 0000,003C,0180,0800,0000,1299  ;2582 =0 J/CAERR   ; NO SET UP ERROR DATA
U 10AD. 0003,003C,0180,0A80,0000,1296  ;2583  J/ENDLP,R[0]_0  ; TAG OK. RESET EXPECTING C/A FLAG
U 1296. 001B,0038,1D80,0800,0010,1297  ;2584 ENDLP: ALU_SC,CLK.UBCC  ; DONE YET?
U 1297. 0000,013C,0180,0A68,0000,10AE  ;2585  Z?,LAB_R[0D]  ; LATCH NUMBER OF EXPECTED CA'S
U 10AE. 0000,003C,0180,0800,0000,128D  ;2586 =0 J/SRLP
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0097
ESKAR3E.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:28                   Page   8

U 10AF, 000D,2000,0180,0800,0010,1298  ;2587  ALU_Q-LB,CLK.UBCC ;  CORRECT NUMBER OF C/A'S?
U 1298, 0000,013C,0180,0800,0000,10B0  ;2588  Z?
U 10B0, 0000,003C,0180,0800,0000,1299  ;2589 =0 J/CAERR
U 10B1, 0000,003E,0180,0800,0000,0002  ;2590  RETURN2    ; OK, CORRECT NUMBER OF GOOD C/A'S
U 1299, 0000,003C,65F0,2C00,0000,129A  ;2591 CAERR: Q_ID[SBI.ERR]
U 129A, 0001,203C,6DF0,2D80,0000,129B  ;2592  RC[0]_Q,Q_ID[FAULT]
U 129B, 0001,203E,0180,0988,0000,0001  ;2593  RC[1]_Q,RETURN1
            ;2594
```

```
                ;2595   .PAGE
                ;2596   .TOC "    TYPE DEVICE NAME"
                ;2597  ;*****************************************************************
                ;2598  ;*
                ;2599  ;       THIS SUBROUTINE IS USED TO SET UP THE D REGISTER
                ;2600  ; WHEN CALLING THE CONSOLE TO TYPE OUT THE NAME OF A DEVICE
                ;2601  ; THAT HAS BEEN FOUND ON THE SBI.
                ;2602  ;
                ;2603  ; WHEN CALLING THE ROUTINE:
                ;2604  ;  THE CONFIGURATION/STATUS REGISTER MUST BE IN D.
                ;2605  ;
                ;2606  ;  THE TR LEVEL MUST BE IN RC[0D]
                ;2607  ;
                ;2608  ; THIS ROUTINE USES BITS <7:3> OF THE CONFIGURATION REGISTER TO
                ;2609  ; DETERMINE THE TYPE OF DEVICE. THESE 5 BITS ARE CURRENTLY
                ;2610  ; DECODED AS FOLLOWS:
                ;2611  ;
                ;2612  ; BIT 7 = 1    - UNKNOWN DEVICE
                ;2613  ; BIT 7 = 0 BITS<6:3>
                ;2614  ;      0   - UNKNOWN DEVICE
                ;2615  ;      1   - MS780-A
                ;2616  ;      2   - MS780-C
                ;2617  ;      3   - UNKNOWN DEVICE
                ;2618  ;      4   - RH780
                ;2619  ;      5   - DW780
                ;2620  ;      6   - DR780
                ;2621  ;      7   - XX780
                ;2622  ;      8   - MA780
                ;2623  ;    9,A,B  - UNKNOWN DEVICE
                ;2624  ;      C   - MS780-E ERROR
                ;2625  ;      D   - MS780-E
                ;2626  ;      E   - MS780-H
                ;2627  ;      F   - MS780-E ERROR
                ;2628  ;
                ;2629  ; NOTE: THE ROUTINE LEAVES THE D-REGISTER SET UP WITH THE DEVICE
                ;2630  ;  CODE IN BITS<7:0>.
                ;2631  ;
                ;2632  ; R[4],R[3],RC[0E],Q,SC,FE,STATE GET CLOBBERED
                ;2633  ;-
                ;2634  ;*****************************************************************
U 129C, 0001,003C,01F8,0A98,0082,129D   ;2635 CONFIG: SC_D,Q_0,R[3]_D   ; SETUP TO BRANCH ON CONFIGURATION
U 129D, 0810,0038,0180,0968,0100,129E   ;2636  FE_SC,D_RC[0D]   ; REGISTER<7:3> AND PUT TR NUMBER
U 129E, 0001,003C,0180,09F0,1400,729F   ;2637  STATE_FE,RC[0E]_D ; IN RC[0E]
U 129F, 0000,163C,0180,0800,0000,1037   ;2638  STATE(7)?   ; VALID CONFIGURATION?
U 1037, 0000,003C,0180,0800,0000,12A0   ;2639 =0111 J/CONFIG_OK_1   ; BRANCH IF YES
U 103F, 0000,003C,0180,0800,0000,10B2   ;2640  NOP
U 10B2, 0818,0039,0D80,0AA0,0000,12B1   ;2641 =0 R[4]_K[.3],TYPE_MSG_3 ; UNKNOWN DEVICE TYPE
U 10B3, 0000,003C,0180,0800,0000,12B0   ;2642  J/CONFIG_EXIT   ; EXIT
                ;2643
                ;2644 CONFIG_OK_1:
U 12A0, 0000,003C,0180,0800,1400,92A1   ;2645  STATE_STATE+FE   ; SETUP TO BRANCH ON CNFG REG BITS<6:3>
U 12A1, 0000,163C,0180,0800,0084,7060   ;2646  STATE?-4?,SC_K[.8] ; DO THE BRANCH AND SETUP SC TO SHIFT D REG
U 1060, 0818,0038,0D80,0800,0000,10B4   ;2647 =0000 D_K[.3],J/TYPE_IT ; UNKNOWN DEVICE TYPE
U 1061, 0818,0038,0580,0800,0000,10B6   ;2648  D_K[.1],J/SIZE_MS780_A ; MS780-A
U 1062, 0818,0038,0980,0800,0000,10B6   ;2649  D_K[.2],J/SIZE_MS780_C ; MS780-C
```

```
U 1063, 0818,0038,0D80,0800,0000,10B4  ;2650  D_K[.3],J/TYPE_IT ; UNKNOWN
U 1064, 0818,0038,1180,0800,0000,10B4  ;2651  D_K[.4],J/TYPE_IT ; RH780
        ;2652  D_ALU.LEFT,SI/7,
U 1065, 0838,0038,0B80,0800,0000,10B4  ;2653      ALU_K[.2],J/TYPE_IT ; DW780
U 1066, 0818,0038,D580,0800,0000,10B4  ;2654  D_K[.6],J/TYPE_IT ; DR780
U 1067, 0818,0038,5D80,0800,0000,10B4  ;2655  D_K[.7],J/TYPE_IT ; ??780
U 1068, 0818,0038,0180,0800,0000,10BA  ;2656  D_K[.8],J/SIZE_MA780 ; MA780
U 1069, 0818,0038,0D80,0800,0000,10B4  ;2657  D_K[.3],J/TYPE_IT ; UNKNOWN
U 106A, 0818,0038,0D80,0800,0000,10B4  ;2658  D_K[.3],J/TYPE_IT ; UNKNOWN
U 106B, 0818,0038,0D80,0800,0000,10B4  ;2659  D_K[.3],J/TYPE_IT ; UNKNOWN
U 106C, 0818,0038,D980,0800,0000,10C4  ;2660  D_K[.9],J/MS780_ERROR ; MS780-E ERROR
U 106D, 0818,0038,F580,0800,0000,10C6  ;2661  D_K[.A],J/SIZE_MS780_E ; MS780-E
        ;2662  D_ALU.LEFT,SI/7,
        ;2663      ALU_0+K[.4]+1,
U 106E, 083B,0010,1380,0800,0000,10C6  ;2664      J/SIZE_MS780_H ; MS780-H
U 106F, 0818,0038,D980,0800,0000,10C4  ;2665  D_K[.9],J/MS780_ERROR ; MS780-E ERROR
        ;2666
        ;2667 ;++
        ;2668 ; THIS ENTRY POINT, FROM THE ABOVE DECISION TREE, IS USED TO
        ;2669 ; TYPE THE MESSAGE SPECIFIED BY THE D REGISTER BITS<7:0> FOLLOWED
        ;2670 ; BY THE TR NUMBER OF THE DEVICE CONTAINED IN RC[0E].
        ;2671 ;--
        ;2672 =0
        ;2673 TYPE_IT:
U 10B4, 0000,003D,0180,0800,0000,10EC  ;2674  CALL[TYPE_DEV_NAME] ; TYPE THE DEVICE NAME
U 10B5, 0000,003C,0180,0800,0000,12B0  ;2675  J/CONFIG_EXIT  ; EXIT
        ;2676 ;
        ;2677 ; THIS ENTRY POINT IS USED TO TYPE THE MS780-A DEVICE MESSAGE FOLLOWED
        ;2678 ; BY THE SIZE OF THE MEMORY IN BYTES.
        ;2679 ;
        ;2680 =0
        ;2681 SIZE_MS780_A:
        ;2682 SIZE_MS780_C:
U 10B6, 0000,003D,0180,0800,0000,10EC  ;2683  CALL[TYPE_DEV_NAME] ; TYPE THE DEVICE NAME AND TR
U 10B7, 0000,003C,D5F8,0800,0084,70B8  ;2684  SC_K[.6],Q_0  ; SETUP FOR LEFT SHIFT
U 10B8, 0000,003D,0180,0800,0000,12BD  ;2685 =0 CALL[DECFLG]  ; DECREMENT TYPEOUT FLAG
U 10B9, 0818,0034,4D80,0A18,0000,12A2  ;2686  D_R[3].AND.K[.FF00] ; GET NUMBER OF ARRAYS PRESENT FIELD
U 12A2, 0D00,003C,4580,0800,0000,12A3  ;2687  D_D+L.SC,K[.8000] ; SHIFT ONE BIT LESS THAN NECESSARY
U 12A3, 0819,0014,4580,0800,0000,12A4  ;2688  D_D+K[.8000]  ; ACCOUNT FOR FIRST ARRAY
U 12A4, 0500,003C,0180,0800,0000,12A5  ;2689  D_D.LEFT  ; SHIFT LAST TIME TO MAKE BYTE ADDRESS
U 12A5, 0001,003C,0180,09F0,0000,10E6  ;2690  RC[0E]_D,J/SIZE_COMMON ; SAVE AND PRINT IT
        ;2691 ;
        ;2692 ; THIS ROUTINE IS USED TO CALCULATE THE MAX BYTE ADDRESS OF AN
        ;2693 ; MA780 MEMORY AND PRINT THE DEVICE NAME/TR MESSAGE FOLLOWED BY
        ;2694 ; THE ARRAY SIZE IN BYTES.
        ;2695 =0
        ;2696 SIZE_MA780:
U 10BA, 0000,003D,0180,0800,0000,10EC  ;2697  CALL[TYPE_DEV_NAME] ; TYPE THE DEVICE NAME AND TR
U 10BB, 0000,003C,0180,0800,0000,10BC  ;2698  NOP
U 10BC, 0000,003D,0180,0800,0000,12BD  ;2699 =0 CALL[DECFLG]  ; DECREMENT THE TYPEOUT FLAG
U 10BD, 0000,003C,0180,0800,0000,10BE  ;2700  NOP
U 10BE, 0838,0039,5D80,0800,0000,12B1  ;2701 =0 TYPE_MSG_E  ; TYPE "  PORT NO= " MESSAGE
U 10BF, 0000,003C,0180,0800,0000,10C0  ;2702  NOP
U 10C0, 0000,003D,0180,0800,0000,12BD  ;2703 =0 CALL[DECFLG]
U 10C1, 0800,003C,0180,0A18,0000,12A6  ;2704  D_R[3]   ; GET CONFIG REG AGAIN
```

```
            ;2705  RC[0E]_ALU,
U 12A6, 0019,0034,0D80,09F0,0000,10C2  ;2706       ALU_D.AND.K[.3] ; GET THE PORT NUMBER OF THE MA780
U 10C2, 0000,003D,0180,0800,0000,12BC  ;2707 =0 TYP_RC[0E]_CRLF  ; TYPE THE PORT NUMBER
U 10C3, 0000,003C,01B0,0800,0000,12B0  ;2708  J/CONFIG_EXIT
            ;2709 ;
            ;2710 ; THIS ROUTINE IS USED TO PRINT THE MS780-E ERROR MESSAGE
            ;2711 ;
            ;2712 =0
            ;2713 MS780_ERROR:
U 10C4, 0000,003D,0180,0800,0000,10EC  ;2714  CALL[TYPE_DEV_NAME] ; TYPE THE DEVICE NAME
U 10C5, 0000,003C,0180,0800,0000,12B0  ;2715  J/CONFIG_EXIT
            ;2716 ;
            ;2717 ; THIS ROUTINE IS USED TO PRINT THE MS780-E MESSAGE AND ARRAY
            ;2718 ; SIZE IN BYTES.
            ;2719 ;
            ;2720 =0
            ;2721 SIZE_MS780_E:
            ;2722 SIZE_MS780_H:
U 10C6, 0000,003D,0180,0800,0000,10EC  ;2723  CALL[TYPE_DEV_NAME] ; PRINT THE DEVICE NAME
U 10C7, 0810,0038,0180,0970,0000,12A7  ;2724  D_RC[0E]
            ;2725  R[6]_D,
U 12A7, 0001,003C,91F0,2EB0,0000,12A8  ;2726   Q_ID[24] ; Get TR Level
U 12A8, 0C00,003C,0180,0800,0000,10C8  ;2727  D_Q
U 10C8, 0000,193D,0180,0800,0000,1010  ;2728 =0 CALL[TR0],D3-0?  ; Generate I/O Base address
U 10C9, 0001,003C,0180,09F0,0000,10CA  ;2729  RC[0E]_D  ; Save I/O Base in RC[0E]
U 10CA, 0000,003D,0180,0800,0000,12BD  ;2730 =0 CALL[DECFLG]  ; Decrement Type-Out Flag
U 10CB, 0000,003C,0180,0800,0000,10CC  ;2731  NOP
U 10CC, 0838,0039,D980,0800,0000,12B1  ;2732 =0 TYPE_MSG_12  ; Print "LOWER CNTRLR " message
U 10CD, 0000,003C,0180,0800,0000,1020  ;2733  NOP
            ;2734 =00 D_0,  ;...
U 1020, 0F00,003D,0180,0800,0000,1236  ;2735     CALL[SELECT.CNTRLR] ; Try to enable the Lower Controller
U 1021, 0000,003C,0180,0800,0000,10D0  ;2736  J/MSE.LOWER.MISCONFIG ; Lower Controller Misconfigured
U 1022, 0000,003D,0180,0800,0000,1246  ;2737 =0 CALL[GET.MEMORY.SIZE] ; Get size of memory on this Controller
U 1023, 0000,003C,0180,0800,0000,10CE  ;2738  NOP
U 10CE, 0000,003D,0180,0800,0000,12A9  ;2739 =0 CALL[MSE.TYP.SIZ] ; Type size of MS780-E/H memory
U 10CF, 0000,003C,0180,0800,0000,10D4  ;2740  J/TRY.UPPER  ; Go and see if there is an upper Controller
            ;2741 =0
            ;2742 MSE.LOWER.MISCONFIG:
U 10D0, 0000,003D,0180,0800,0000,12BD  ;2743  CALL[DECFLG]  ; Decrement Type-Out Flag
U 10D1, 0000,003C,0180,0800,0000,10D2  ;2744  NOP
U 10D2, 0818,0039,2180,0800,0000,12B9  ;2745 =0 TYP_MSG_14_CRLF  ; Print Misconfigured message
U 10D3, 0000,003C,0180,0800,0000,10D4  ;2746  NOP
            ;2747 =0
            ;2748 TRY.UPPER:
U 10D4, 0000,003D,0180,0800,0000,12BD  ;2749  CALL[DECFLG]  ; Decrement Type-Out Flag
U 10D5, 0000,003C,0180,0800,0000,10D6  ;2750  NOP
U 10D6, 0838,0039,DB80,0800,0000,12B1  ;2751 =0 TYPE_MSG_13  ; Print "UPPER CNTRLR " message
U 10D7, 0000,003C,0180,0800,0000,1024  ;2752  NOP
            ;2753 =00 D_K[.2],  ;
U 1024, 0818,0039,0980,0800,0000,1236  ;2754     CALL[SELECT.CNTRLR] ; Try to select the upper controller
U 1025, 0000,003C,0180,0800,0000,10DA  ;2755  J/MSE.UPPER.MISCONFIG  ; Upper Controller Misconfigured
U 1026, 0000,003D,0180,0800,0000,1246  ;2756 =0 CALL[GET.MEMORY.SIZE] ; Get the size of memory on the Upper Cntrlr
U 1027, 0000,003C,0180,0800,0000,10D8  ;2757  NOP
U 10D8, 0000,003D,0180,0800,0000,12A9  ;2758 =0 CALL[MSE.TYP.SIZ] ; Type size of MS780-E/H memory
U 10D9, 0000,003C,0180,0800,0000,12AE  ;2759  J/MSE.CNFG.EXIT  ; Exit CONFIG subroutine
```

```
            ;2760 =0
            ;2761 MSE.UPPER.MISCONFIG:
U 10DA, 0000.003D.0180.0800.0000.12BD  ;2762  CALL[DECFLG]  ; Decrement Type-Out flag
U 10DB, 0000.003C.0180.0800.0000.10DC  ;2763  NOP
U 10DC, 0818.0039.2180.0800.0000.12B9  ;2764 =0 TYP_MSG_14_CRLF  ; Print Misconfigured message
U 10DD, 0000.003C.0180.0800.0000.12AE  ;2765  J/MSE.CNFG.EXIT  ; Exit CONFIG subroutine
            ;2766 ;
            ;2767 ; * * *
```

```
            ;2807  .PAGE
            ;2808  .TOC "   TYPE DEV NAME AND TR MESSAGE"
            ;2809 ;**
            ;2810 ;
            ;2811 ; THIS SUBROUTINE IS USED TO TYPE THE MESSAGE NUMBER CONTAINED IN
            ;2812 ; D REGISTER BITS<7:0>. UPON ENTRY, THE SC MUST CONTAIN 8 AND THE Q REGISTER
            ;2813 ; MUST BE CLEAR. UPON EXIT, THE D REGISTER IS UNMODIFIED AND R[4] CONTAINS
            ;2814 ; A COPY OF THE D REGISTER.
            ;2815 ;
            ;2816 ;--
            ;2817 =0
            ;2818 TYPE_DEV_NAME:
U 10EC, 0001,003D,0180,0AA0,0000,12B1  ;2819  R[4]_D,CALL[MSGCOM] ; TYPE THE DEVICE NAME
U 10ED, 0000,003C,0180,0800,0000,10EE  ;2820  NOP
U 10EE, 0000,003D,0180,0800,0000,12BD  ;2821 =0 CALL[DECFLG]
U 10EF, 0000,003C,0180,0800,0000,10F0  ;2822  NOP
U 10F0, 0818,0039,B580,0800,0000,12B1  ;2823 =0 TYPE_MSG_C  ; TYPE " AT TR "
U 10F1, 0000,003C,0180,0800,0000,10F2  ;2824  NOP
U 10F2, 0000,003D,0180,0800,0000,12BD  ;2825 =0 CALL[DECFLG]
U 10F3, 0000,003C,0180,0800,0000,10F4  ;2826  NOP
U 10F4, 0000,003D,0180,0800,0000,12BC  ;2827 =0 TYP_RC[0E]_CRLF  ; TYPE THE TR LEVEL
U 10F5, 0800,003E,0180,0A20,0000,0001  ;2828  D_R[4],RETURN1  ; EXIT
            ;2829
```

```
            ;2830  .PAGE
            ;2831  .TOC "   PRINT ASCII"
            ;2832 ;*****************************************************************
            ;2833 ;*
            ;2834 ; THIS ROUTINE CHECKS ID[TO] (THE TYPE OUT FLAG).
            ;2835 ; IF ID[TO] IS NEGATIVE, THE MESSAGE IS TYPED, AND ID[TO]
            ;2836 ; IS INCREMENTED.  THIS IS THE COMMUNICATION DEVICE BETWEEN
            ;2837 ; ESKAE (THE MICRO TEST MONITOR), AND THE GO CHAIN OVERLAYS
            ;2838 ; TO TELL WHETHER LOOP ON TEST OR SECTION HAS BEEN SPECIFIED.
            ;2839 ;
            ;2840 ; MSGCOM: IS USED BY A MACRO CALL "TYP[E]_MSG_X[_CRLF] WHERE D STILL
            ;2841 ;  NEEDS TO BE SHIFTED 9 PLACES.
            ;2842 ; MSGCOM_1:
            ;2843 ;  ENTERED WITH SC CONTAINING THE CODE FOR D REGISTER BYTE 0.
            ;2844 ; CNFCOM: IS USED BY THE SUBROUTINE "CONFIG" WHERE D DOES
            ;2845 ;  NOT NEED ANY FURTHER SHIFTING.
            ;2846 ;-
            ;2847 ;*****************************************************************
U 12B1, 0000,003C,F580,0800,0084,72B2  ;2848 MSGCOM: SC_K[.A]
            ;2849 MSGCOM_1:
U 12B2, 0000,003C,0180,0800,0100,12B3  ;2850  FE_SC
U 12B3, 0000,003C,D9F8,0800,0084,72B4  ;2851  SC_K[.9],Q_0
U 12B4, 0D00,003C,0180,0800,0000,12B5  ;2852  D_DAL.SC
U 12B5, 0000,003C,C1F0,2C00,0000,12B6  ;2853 CNFCOM: Q_ID[TO]
U 12B6, 0C01,203C,01E0,0800,0010,12B7  ;2854  ALU_Q,CLK.UBCC,Q_D,D_Q
U 12B7, 0819,1B14,0580,0800,0000,1047  ;2855  ALU.N?,D_D+K[.1]
U 1047, 0C00,003E,0180,0800,0000,0001  ;2856 =0111 RETURN1,D_Q
U 104F, 0C00,003C,C180,3C00,0081,12B8  ;2857  ID[TO]_D,D_Q,SC_FE
U 12B8, 0819,0030,1D80,0800,0000,105E  ;2858  D_D.OR.K[SC],J/CALLCON
            ;2859
            ;2860 MSGCOM_CRLF:
U 12B9, 0018,0038,65C0,0800,0000,12BA  ;2861  Q_K[.10]
U 12BA, 0019,2030,0980,0800,0082,12B2  ;2862  SC_Q.OR.K[.2],J/MSGCOM_1
U 12BB, 0000,003C,6580,0800,0084,72B2  ;2863 RCECOM: SC_K[.10],J/MSGCOM_1
            ;2864 RCECOM_CRLF:
U 12BC, 0000,003C,2180,0800,0084,72B2  ;2865  SC_K[.14],J/MSGCOM_1
            ;2866
```

```
            ;2867  .PAGE
            ;2868  .TOC "   DECREMENT TYPE-OUT FLAG"
            ;2869 ;*************************************************************
            ;2870 ;*
            ;2871 ; THIS ROUTINE DECREMENTS THE TYPE OUT FLAG ID[T0]
            ;2872 ;-
            ;2873 ;*************************************************************
U 12BD, 0000,003C,C1F0,2C00,0000,12BE  ;2874 DECFLG: Q_ID[T0]
U 12BE, 0819,2000,0580,0800,0000,12BF  ;2875  D_Q-K[.1]
U 12BF, 0000,003E,C180,3C00,0000,0001  ;2876  ID[T0]_D,RETURN1
            ;2877
```

```
            ;2878  .PAGE
            ;2879  .TOC "   SBIDAT (FOR NEXUS RESPONSE TEST"
            ;2880 ;*************************************************************
            ;2881 ;+
            ;2882 ; THIS ROUTINE GATHERS ERROR DATA FOR THE
            ;2883 ; SBI NEXUS RESPONSE TEST.
            ;2884 ;-
U 12C0. 0000.003C.91F0.2C00.0000.12C1   ;2885 SBIDAT: Q_ID[24]
U 12C1. 0C00.003C.0180.0800.0000.10F6   ;2886  D_Q
U 10F6. 0000.193D.0180.0800.0000.1010   ;2887 =0 CALL[TR0],D3-0?
U 10F7. 0001.003C.95F0.2DF0.0000.12C2   ;2888  RC[0E]_D,Q_ID[25]
U 12C2. 0001.203C.35F0.2DE8.0000.12C3   ;2889  RC[0D]_Q,Q_ID[VECTOR]
U 12C3. 0001.203C.65F0.2DD8.0000.12C4   ;2890  RC[0B]_Q,Q_ID[SBI.ERR]
U 12C4. 0001.203C.6DF0.2DD0.0000.12C5   ;2891  RC[0A]_Q,Q_ID[FAULT]
U 12C5. 0001.203E.0180.09C8.0000.0001   ;2892  RC[9]_Q,RETURN1
            ;2893
```

```
            ;2894  .PAGE
            ;2895  .TOC "   FIND CPU ID FROM SILO"
            ;2896 ;////////////////////////////////////////////////////////////////
            ;2897 ;+
            ;2898 ;       THIS ROUTINE IS USED TO GET THE CPU'S ID OUT OF THE
            ;2899 ; SILO AND **RETURN WITH THAT ID IN D<27:23>**
            ;2900 ;
            ;2901 ;    THIS IS DONE SO THAT THE TEST MAY PUT THAT ID DIRECTLY
            ;2902 ; INTO ID[MAINT] OR DO WHATEVER ELSE IT PLEASES.
            ;2903 ;
            ;2904 ; **THIS ROUTINE MUST NOT BE CALLED BEFORE
            ;2905 ; "CPU ID TEST" HAS BEEN PERFORMED. **
            ;2906 ;
            ;2907 ; ****THIS ROUTINE CLOBBERS THE WHOLE WORLD****.
            ;2908 ;
            ;2909 ;  SC,R[0F],ID[COMP],THE WHOLE SILO, THE VA, Q,
            ;2910 ;  ALL OF THE RC SCRATCH PADS (EXCEPT RC[0F])
            ;2911 ;  AND SBI ERROR  (CLEARED).
            ;2912 ;
            ;2913 ;-
            ;2914 =0
U 10F8, 0000.003D.0180.0800.0000.1092  ;2915 MNTID: CALL[CPUCA]
U 10F9, 0000.003C.C580.3C00.0000.12C6  ;2916  ID[31]_D  ; SAVE C/A PATTERN
U 12C6, 0838.0038.8D80.0800.0000.12C7  ;2917  D_ALU.LEFT,SI/ZERO,ALU_K[.1F]
U 12C7, 0B00.003C.C5F0.2C00.0000.12C8  ;2918  D_D.SWAP,Q_ID[31]
U 12C8, 089D.0036.0180.0800.0000.0001  ;2919  D_ALU.RIGHT2,SI/ZERO,ALU_D.AND.Q,RETURN1
            ;2920
```

```
            ;2921   .PAGE
            ;2922   .TOC "    SILO LOCK PATTERN FOR RD DATA"
            ;2923 ;////////////////////////////////////////////////////////////////
            ;2924 ;*
            ;2925 ;       THIS ROUTINE GENERATES THE PATTERN THAT WILL LOCK
            ;2926 ; UP THE SILO ON THE FIRST OCCURRENCE OF READ DATA
            ;2927 ;
            ;2928 ; REMEMBER THAT THE ID YOU WANT TO LOCK ON MUST BE IN
            ;2929 ; ID[MAINT].
            ;2930 ;
            ;2931 ; D IS RETURNED WITH THE VALUE 100E0000.
            ;2932 ;-
U 12C9.  0838.0038.5D80.0800.0000.12CA   ;2933 LK1RDT: D_ALU.LEFT,SI/ZERO,ALU_K[.7]
U 12CA.  0000.003C.01F8.0800.0084.72CB   ;2934  SC_K[.8],Q_0
U 12CB.  0D00.003C.0180.0800.0000.12CC   ;2935  D_DAL.SC
U 12CC.  0819.0030.6580.0800.0000.12CD   ;2936  D_D.OR.K[.10]
U 12CD.  0B00.003E.0180.0800.0000.0001   ;2937  D_D.SWAP,RETURN1
            ;2938
```

```
                ;2939  .PAGE "TEST 17B SBI.HOLD TEST"
                ;2940 ;*****************************************************************
                ;2941 ;**
                ;2942 ; TEST 17B SBI.HOLD TEST
                ;2943 ;
                ;2944 ; TEST DESCRIPTION:
                ;2945 ;
                ;2946 ;  THIS TEST CHECKS THAT HOLD, AND HOLD+UNJAM GO OUT
                ;2947 ; ON THE SBI PROPERLY FROM THE CPU.
                ;2948 ;
                ;2949 ; LOGIC DESCRIPTION:
                ;2950 ;
                ;2951 ;  LOGIC ON SBH,SBL, AND THE TERMINATOR IS
                ;2952 ; CHECKED HERE.
                ;2953 ;
                ;2954 ; ERROR DESCRIPTION:
                ;2955 ;
                ;2956 ; THE LAST 13 SILO LOCATIONS, PLUS ID[FAULT], AND ID[SBI.ERR]
                ;2957 ;
                ;2958 ; SUBTEST 1
                ;2959 ;  NULL
                ;2960 ;  NULL
                ;2961 ;  11 LOCATIONS OF HOLD=00000001
                ;2962 ;  ID[FAULT] (ID ADRS:1B)
                ;2963 ;  ID[SBI.ERR] (ID ADRS:19)
                ;2964 ;
                ;2965 ; SUBTEST 2
                ;2966 ;  NULL
                ;2967 ;  NULL
                ;2968 ;  11 LOCATIONS OF HOLD+UNJAM = 00000001
                ;2969 ;  ID[FAULT]
                ;2970 ;  ID[SBI.ERR]
                ;2971 ;--
                ;2972 ;*****************************************************************
                ;2973 =0
U 10FA, 0000,003D,0180,0800,0000,1216  ;2974 HLDTST: NEWTST
U 10FB, 0018,0038,6180,09F8,0000,12CE  ;2975  RC[0F]_K[.F]
U 12CE, 0000,003C,8980,0800,0084,72CF  ;2976  SC_K[.D]
U 12CF, 0003,001C,0180,0AF8,0000,10FC  ;2977  ALU_NOT.MASK,R[0F]_ALU ; SET BIT FOR TIMEOUT TRAP
                ;2978 ;////////////////////////////////////////////////////////////////
                ;2979 ;*
                ;2980 ; SUBTEST 1 MCT/SBI.HOLD
                ;2981 ;-
U 10FC, 0000,003D,0180,0800,0000,122D  ;2982 =0 SUBTEST
U 10FD, 0000,003C,0180,0800,0000,10FE  ;2983  NOP
U 10FE, 0000,003D,0180,0800,0000,1228  ;2984 =0 ERLOOP
U 10FF, 0000,003C,0180,0800,0000,110A  ;2985  NOP
U 110A, 0000,003D,0180,0800,0000,1280  ;2986 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 110B, 0000,003C,0180,0800,0000,1110  ;2987  NOP
U 1110, 0000,003D,0180,0800,0000,1010  ;2988 =0 CALL,J/TR0  ; D GETS 20000000
U 1111, 0001,003C,7180,3C00,0200,1112  ;2989  VA_D,ID[COMP]_D  ; LOCK SILO AFTER 16 CYCLES
U 1112, 0F00,003D,0180,8000,0000,126C  ;2990 =0 CALL,J/UNJAM2,D_0,MCT/SBI.HOLD ; DO REFERENCE,PUT SILO IN RC'S
U 1113, 0000,003C,1980,0800,1404,7114  ;2991  STATE_K[ZERO]  ; ONE PATTERN EXPECTED FLAG FOR SILRTN
U 1114, 0000,003D,0180,0800,0000,10A0  ;2992 =0 CALL,J/SLOLCK
U 1115, 0818,0038,0580,0800,0000,12D0  ;2993  D_K[.1]  ; EXPECTED DATA FOR SILO ROUTINE
```

```
U 12D0. 0018.0038.8980.0AE8.0000.1028  ;2994  R[0D]_K[.D]  ; NUMBER OF PATTERN REPETITIONS FOR SILRTN
U 1028. 0000.003D.0180.0800.0000.128C  ;2995 =00 CALL.J/SILRTN
U 1029. 0000.003D.0180.0800.0000.1229  ;2996  ERROR1   ; SILO DATA INCORRECT
U 102A. 0000.003C.0180.0800.0000.1116  ;2997  NOP
        ;2998 =
        ;2999 ;////////////////////////////////////////////////////////////////
        ;3000 ;*
        ;3001 ; SUBTEST 2 MCT/SBI.HOLD*UNJAM
        ;3002 ;-
U 1116. 0000.003D.0180.0800.0000.122D  ;3003 =0 SUBTEST
U 1117. 0000.003C.0180.0800.0000.1118  ;3004  NOP
U 1118. 0000.003D.0180.0800.0000.1228  ;3005 =0 ERLOOP
U 1119. 0000.003C.1980.0800.1404.711A  ;3006  STATE_K[ZERO]  ; ONE PATTERN EXPECTED FLAG FOR SILRTN
U 111A. 0000.003D.0180.0800.0000.1280  ;3007 =0 CALL.J/SILCLR  ; CLEAR OUT SILO
U 111B. 0000.003C.0180.0800.0000.111C  ;3008  NOP
U 111C. 0000.003D.0180.0800.0000.1010  ;3009 =0 CALL.J/TR0  ; D GETS 20000000
U 111D. 0001.003C.7180.3C00.0200.111E  ;3010  VA_D.ID[COMP]_D  ; LOCK SILO AFTER 16 CYCLES
U 111E. 0F00.003D.0180.8800.0000.1269  ;3011 =0 CALL.J/UNJAM1,D_0,MCT/SBI.HOLD*UNJAM ; DO THE REFERENCE
U 111F. 0000.003C.0180.0800.0000.1120  ;3012  NOP
U 1120. 0000.003D.0180.0800.0000.10A0  ;3013 =0 CALL.J/SLOLCK
U 1121. 0018.0038.8980.0AE8.0000.1030  ;3014  R[0D]_K[.D]  ; NUMBER OF REPETITIONS EXPECTED
U 1030. 0818.0039.0580.0800.0000.128C  ;3015 =00 CALL.J/SILRTN,D_K[.1]
U 1031. 0000.003D.0180.0800.0000.1229  ;3016  ERROR1
U 1032. 0000.003C.0180.0800.0000.1122  ;3017  NOP
        ;3018 =
```

```
;3019  .PAGE "TEST 17C TIMEOUT TEST"
;3020 ;*****************************************************************
;3021 ;**
;3022 ; TEST 17C TIMEOUT TEST
;3023 ;
;3024 ; TEST DESCRIPTION:
;3025 ;
;3026 ;      SUBTEST 1 DOES A WRITE REFERENCE TO A NON-EXISTENT
;3027 ; I/O ADDRESS, EXPECTING BOTH A TIME OUT TRAP, AND A TIME OUT
;3028 ; INTERRUPT. (WRITES TO I/O CAUSE BOTH, NORMAL WRITES CAUSE ONLY
;3029 ; INTERRUPTS).
;3030 ;      ERROR CALLS ARE MADE IF:
;3031 ;  NO TRAP OCCURRS
;3032 ;  NO INTERRUPT OCCURRS
;3033 ;  THE WRONG INTERRUPT VECTOR IS GENERATED.
;3034 ;      SUBTEST 2 DOES A READ REFERENCE TO A NON-EXISTENT I/O
;3035 ; ADDRESS, EXPECTING A TIMEOUT TRAP
;3036 ;
;3037 ; LOGIC DESCRIPTION:
;3038 ;
;3039 ;      LOGIC ON SBH,SBL,CEH, AND ICL ARE TESTED HERE
;3040 ;
;3041 ; DATA DESCRIPTION:
;3042 ; *SUBTEST 1*
;3043 ;
;3044 ; VA REFERENCED
;3045 ; ERROR CODE: (FOR FAILCHAIN)
;3046 ;  0=>NO TRAP OCCURRED
;3047 ;  1=>NO INTERRUPT OR WRONG VECTOR
;3048 ; UNDEFINED
;3049 ; UNDEFINED
;3050 ; UNDEFINED
;3051 ; UNDEFINED
;3052 ; INTERRUPT VECTOR REGISTER (ID ADRS:0D)
;3053 ; SBI ERROR REGISTER (ID ADRS:19)
;3054 ; FAULT REGISTER (ID ADRS:18)
;3055 ;
;3056 ; *SUBTEST 2*
;3057 ;
;3058 ; VA REFERENCED
;3059 ; DATA READ (IF NOT = FF, SBI IS MISCONFIGURED) SHOULDN'T BE ANY
;3060 ; SBI ERROR REGISTER
;3061 ; FAULT REGISTER (ID ADRS:1B)
;3062 ;
;3063 ;--
;3064 ;*****************************************************************
;3065 ;/////////////////////////////////////////////////////////////////
;3066 ;*
;3067 ; SUBTEST 1 VA=20000000,WRITE.P
;3068 ;-
;3069 =0
U 1122, 0000.003D.0180.0800.0000.1216  ;3070 TIMO: NEWTST
U 1123, 0018.0038.D980.09F8.0000.1124  ;3071  RC[0F]_K[.9]
U 1124, 0000.003D.0180.0800.0000.122D  ;3072 =0 SUBTEST
U 1125, 0000.003C.0180.0800.0000.1126  ;3073  NOP
```

```
U 1126. 0000.003D.0180.0800.0000.1228  ;3074 =0 ERLOOP
U 1127. 0000.003C.8980.0800.0084.72D1  ;3075  SC_K[.D]
U 12D1. 0003.001C.0180.0AF8.0000.1128  ;3076  R[0F]_ALU,ALU_NOT.MASK  ; SET BIT FOR TIMEOUT TRAP
U 1128. 0000.003D.0180.0800.0000.1266  ;3077 =0 CALL,J/HLD.UNJ ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 1129. 0003.003C.0180.09E8.0000.112A  ;3078  RC[0D]_0
U 112A. 0000.003D.0180.0800.0000.1253  ;3079 =0 CALL,J/ENFLTI  ; ENABLE SBI FAULT INTERRUPTS
U 112B. 0003.003C.0180.09E0.0000.112C  ;3080  RC[0C]_0
U 112C. 0000.003D.0180.0800.0000.1010  ;3081 =0 CALL,J/TR0  ; D GETS 20000000
U 112D. 0001.003C.0180.09F0.0200.12D2  ;3082  RC[0E]_D,VA_D
U 12D2. 0000.003C.0180.A800.0000.12D3  ;3083  FS/MCT,MCT/WRITE.P ; DO THE REFERENCE
U 12D3. 0000.003C.0180.0A78.0010.12D4  ;3084  ALU_R[0F],CLK.UBCC ; SEE IF TRAP OCCURRED
U 12D4. 0000.013C.0180.0800.0000.112E  ;3085  Z?
U 112E. 0000.003D.0180.0800.0000.1229  ;3086 =0 ERROR1   ; NO TRAP OCCURRED
U 112F. 0018.0038.0580.09E8.0000.1038  ;3087  RC[0D]_K[.1]  ; SET ERROR CODE
U 1038. 0000.003D.0180.0800.0000.1262  ;3088 =00 CALL,J/WINTER  ; WAIT 528, STROBE VECTOR
U 1039. 0000.003D.0180.0800.0000.1229  ;3089  ERROR1   ; NO INTERRUPT OCCURRED
U 103A. 0000.003D.0180.0800.0000.1229  ;3090  ERROR1   ; WRONG INTERRUPT VECTOR
U 103B. 0000.003C.0180.0800.0000.1130  ;3091  NOP   ; OK
        ;3092 ;////////////////////////////////////////////////////////////////
        ;3093 ;*
        ;3094 ; SUBTEST 2 VA=20000000, READ.P
        ;3095 ;-
        ;3096 =0
U 1130. 0000.003D.0180.0800.0000.122D  ;3097 RTIMO: SUBTEST
U 1131. 0000.003C.0180.0800.0000.1132  ;3098  NOP
U 1132. 0000.003D.1180.0800.0084.722F  ;3099 =0 SETRCF[.4]
U 1133. 0000.003C.0180.0800.0000.1134  ;3100  NOP
U 1134. 0000.003D.0180.0800.0000.1228  ;3101 =0 ERLOOP
U 1135. 0000.003C.8980.0800.0084.72D5  ;3102  SC_K[.D]
U 12D5. 0003.001C.0180.0AF8.0000.1136  ;3103  R[0F]_ALU,ALU_NOT.MASK ; SET BIT FOR TRAP CATCHER
U 1136. 0000.003D.0180.0800.0000.1266  ;3104 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 1137. 0000.003C.0180.0800.0000.1138  ;3105  NOP
U 1138. 0000.003D.0180.0800.0000.1010  ;3106 =0 CALL,J/TR0  ; D = 20000000
U 1139. 0001.003C.0180.09F0.0200.12D6  ;3107  RC[0E]_D,VA_D
U 12D6. 0818.0038.4980.0800.0000.12D7  ;3108  D_K[.FF]
U 12D7. 0001.003C.0180.09E8.0000.12D8  ;3109  RC[0D]_D
U 12D8. 0000.003C.0180.C800.0000.12D9  ;3110  FS/MCT,MCT/READ.P ; DO THE REFERENCE
U 12D9. 0001.003C.0180.09E8.0000.12DA  ;3111  RC[0D]_D  ; SEE IF A DEVICE ANSWERS
U 12DA. 0000.003C.0180.0A78.0010.12DB  ;3112  ALU_R[0F],CLK.UBCC ; DID A TRAP OCCURR?
U 12DB. 0000.013C.0180.0800.0000.113A  ;3113  Z?
U 113A. 0000.003C.0180.0800.0000.12DC  ;3114 =0 J/NTERR
U 113B. 0000.003C.0180.0800.0000.12DF  ;3115  J/ENDTMO  ; OK
U 12DC. 0000.003C.65F0.2C00.0000.12DD  ;3116 NTERR: Q_ID[SBI.ERR]
U 12DD. 0001.203C.6DF0.2DE0.0000.12DE  ;3117  RC[0C]_Q,Q_ID[FAULT]
U 12DE. 0001.203D.0180.09D8.0000.1229  ;3118  RC[0B]_Q,ERROR1 ; NO TRAP OCCURRED
U 12DF. 0000.003C.0180.0800.0000.113C  ;3119 ENDTMO: NOP
```

```
            ;3120  .PAGE "TEST 17D CPU ID TEST"
            ;3121 ;*****************************************************************
            ;3122 ;**
            ;3123 ; TEST 17D CPU ID TEST
            ;3124 ;
            ;3125 ; TEST DESCRIPTION:
            ;3126 ;
            ;3127 ;      THIS TEST CHECKS THAT THE CPU ID THAT APPEARS IN THE
            ;3128 ; SILO <29:25> IS:
            ;3129 ;  A. CONSISTENT WITH THE SBI TR BIT THAT APPEARS IN
            ;3130 ;     SILO<15:00>, AND
            ;3131 ;  B. CONSISTENT WITH THE MAINTENENCE ID THAT APPEARS
            ;3132 ;     IN ID[MAINT] <27:23>, IN THAT THE CPU SHOULD NOT
            ;3133 ;     DETECT A MULTIPLE TRANSMITTER FAULT WITH IT'S OWN
            ;3134 ;     (SILO<29:25>) ID.
            ;3135 ;
            ;3136 ; LOGIC DESCRIPTION:
            ;3137 ; LOGIC ON SBH,TRS IS CHECKED HERE
            ;3138 ;
            ;3139 ; DATA DESCRIPTION:
            ;3140 ;
            ;3141 ; IF ERROR OCCURS BECAUSE SILO<29:25> #> SILO<15:00>
            ;3142 ;  ERROR CODE (FOR FAILCHAIN) = A(X)
            ;3143 ;  CPU COMMAND ADDRESS FOR A READ.P (AS IT APPEARS IN SILO)
            ;3144 ;
            ;3145 ; IF ERROR OCCURS BECAUSE THE CPU DETECTED MULTIPLE TRANSMITTERS
            ;3146 ; WITH IT'S "OWN" ID:
            ;3147 ;  CPU COMMAND ADDRESS FOR A READ.P (AS IT APPEARS IN SILO)
            ;3148 ;  EXPECTED ID[MAINT]
            ;3149 ;  ID[MAINT] (ID ADRS:1D)
            ;3150 ;  SBI ERROR REGISTER (ID ADRS:19)
            ;3151 ;  FAULT REGISTER (ID ADRS:18)
            ;3152 ;  ERROR CODE (FOR FAILCHAIN) = B(X)
            ;3153 ;--
            ;3154 ;*****************************************************************
            ;3155 =0
U 113C. 0000.003D,0180.0800,0000.1216  ;3156 CPUIDT: NEWTST
U 113D. 0018.0038,0980.09F8,0000.113E  ;3157  RC[0F]_K[.2]
U 113E. 0000.003D,0180.0800,0000.12BD  ;3158 =0 CALL,J/DECFLG  ; DECREMENT TYPE OUT FLAG
U 113F. 0018.0038,F580.09E8,0000.1140  ;3159  RC[0D]_K[.A]
U 1140. 0000.003D,0180.0800,0000.1228  ;3160 =0 ERLOOP
U 1141. 0000.003C,0180.0800,0000.1142  ;3161  NOP
U 1142. 0000.003D,0180.0800,0000.1092  ;3162 =0 CALL,J/CPUCA  ; GET CPU ID FROM SILO (MCT/READ.P)
U 1143. 0001.003C,0180.09F0,0000.12E0  ;3163  RC[0E]_D
U 12E0. 0019.0034,C180.0A80,0000.12E1  ;3164  ALU_D.AND.K[.FFFF],R[0]_ALU ; SAVE SILO <15:00>
U 12E1. 0B00.003C,0180.0800,0000.12E2  ;3165  D_D.SWAP  ; GET ID BITS INTO D<04:00>
U 12E2. 0600.003C,0180.0800,0000.12E3  ;3166  D_D.RIGHT  ;
U 12E3. 0019.0034,8D80.0800,0082.12E4  ;3167  ALU_D.AND.K[.1F],SC_ALU ; SET MASK TO RECREATE SILO<15:00>
U 12E4. 0803.001C,0180.0800,0000.12E5  ;3168  D_NOT.MASK
U 12E5. 0819.0034,C180.0A00,0000.12E6  ;3169  D_D.AND.K[.FFFF],LAB_R[0] ; GET RID OF "TR 16"
U 12E6. 001C.2000,0180.0800,0010.12E7  ;3170  ALU_LA-D,CLK.UBCC
U 12E7. 0000.013C,0180.0800,0000.1144  ;3171  Z?
U 1144. 0000.003D,0180.0800,0000.1229  ;3172 =0 ERROR1  ; SILO INCORRECT
U 1145. 0818.0038,1D80.0980,0000.12E8  ;3173  ALU_SC,D_ALU,RC[0]_ALU ; OK, NOW TRY MULT XMIT
U 12E8. 0018.0038,D580.09F8,0000.12E9  ;3174  RC[0F]_K[.6]
```

```
U 12E9. 0819.0030.7580.0800.0000.12EA  ;3175  D_D.OR.K[.20]
U 12EA. 0000.003C.B980.0800.0084.72EB  ;3176  SC_K[.19]
U 12EB. 0000.003C.09F8.0948.0084.B2EC  ;3177  SC_SC-K[.2],Q_0,LC_RC[9]
U 12EC. 0D13.0010.0180.09C8.0000.12ED  ;3178  D_DAL.SC,ALU_0+LC+1,RC[9]_ALU
U 12ED. 0000.003C.7580.3C00.0000.12EE  ;3179  ID[MAINT]_D  ; FORCE MULT XMIT,CPU ID
U 12EE. 0000.003C.75F0.2C00.0000.12EF  ;3180  Q_ID[MAINT]
U 12EF. 0001.003C.0180.09E8.0000.12F0  ;3181  RC[0D]_D  ; EXPECTED ID MAINT
U 12F0. 0001.203C.0180.09E0.0000.1146  ;3182  RC[0C]_Q  ; GOT ID[MAINT]
U 1146. 0000.003D.0180.0800.0000.1010  ;3183 =0 CALL,J/TR0  ; D_NON EXISTENT I/O ADRS
U 1147. 0001.003C.0180.0800.0200.1148  ;3184  VA_D
U 1148. 0000.003D.0180.0800.0000.1253  ;3185 =0 CALL,J/ENFLTI  ; ENABLE SBI FAULT INTERRUPTS
U 1149. 0000.003C.8980.0800.0084.72F1  ;3186  SC_K[.D]
U 12F1. 0003.001C.0180.0AF8.0000.12F2  ;3187  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 12F2. 0000.003C.0180.A800.0000.12F3  ;3188  MCT/WRITE.P  ; DO THE REFERENCE
U 12F3. 0F00.003C.E580.0800.0084.72F4  ;3189  D_0,SC_K[.1A]  ; MASK FOR ALL BUT FAULT<27:26>
U 12F4. 0801.001C.0180.0800.0080.D2F5  ;3190  D_D.ORNOT.MASK,SC_SC+1 ;...
U 12F5. 0801.001C.6DF0.2C00.0000.12F6  ;3191  D_D.ORNOT.MASK,Q_ID[FAULT]
U 12F6. 001D.0034.0180.0800.0010.12F7  ;3192  ALU_D[AND]Q,CLK.UBCC
U 12F7. 0000.013C.0180.0800.0000.114A  ;3193  Z?
U 114A. 0001.203C.65F0.2DD0.0000.12FC  ;3194 =0 J/ERFLT,RC[0A]_Q,Q_ID[SBI.ERR] ; NO SET DATA MAKE ERROR CALL
U 114B. 0F00.003C.0180.0900.0000.12F8  ;3195  LC_RC[0],D_0  ; PUT CPU TR IN RC[0E],CLEAR MAINT
U 12F8. 0010.0038.7580.3DF0.0000.12F9  ;3196  RC[0E]_LC,ID[MAINT]_D
U 12F9. 081B.0000.0580.0800.0000.12FA  ;3197  D_0-K[.1]
U 12FA. 0801.0028.6580.3C00.0000.12FB  ;3198  ID[SBI.ERR]_D,D_NOT.D ; OK, CLEAR REGISTERS
U 12FB. 0000.003C.6580.3C00.0000.114C  ;3199  ID[SBI.ERR]_D
U 114C. 0818.0039.6180.0800.0000.12B1  ;3200 =0 TYPE_MSG_F  ; TYPE "CPU TR = "
U 114D. 0000.003C.0180.0800.0000.114E  ;3201  NOP
U 114E. 0000.003D.0180.0800.0000.12BD  ;3202 =0 CALL[DECFLG]
U 114F. 0000.003C.0180.0800.0000.1150  ;3203  NOP
U 1150. 0000.003D.0180.0800.0000.12BC  ;3204 =0 TYP_RC[0E]_CRLF  ; TYPE THE DATA
U 1151. 0000.003C.0180.0800.0000.12FD  ;3205  J/ENDCP
U 12FC. 0001.203D.0180.09D8.0000.1229  ;3206 ERFLT: ERROR1,RC[0B]_Q  ; CPU SAW MULT XMIT  FAULT
U 12FD. 0000.003C.0180.0800.0000.1152  ;3207 ENDCP: NOP
```

```
;3208   .PAGE "TEST 17E COMMAND ADDRESS TEST"
;3209  ;*************************************************************
;3210  ;*
;3211  ; TEST 17E COMMAND ADDRESS TEST
;3212  ;
;3213  ; TEST DESCRIPTION:
;3214  ; THIS TEST DOES MCT REFERENCES TO NON-EXISTENT ADDRESSES
;3215  ; IN ORDER TO EXAMINE THE SILO AND CHECK THAT COMMAND ADDRESS
;3216  ; GOES OUT ON THE SBI PROPERLY FROM THE CPU.
;3217  ;
;3218  ; LOGIC DESCRIPTION:
;3219  ; LOGIC ON SBH AND SBL ARE CHECKED HERE
;3220  ; IF A DEVICE IS CORRUPTING THE SBI WITH BAD PARITY, OR
;3221  ; MULTIPLE TRANSMITTERS, IT WILL BE CAUGHT IN THIS TEST.
;3222  ; ALSO IF THE SBI IS MISCONFIGURED (AND A DEVICE ANSWERS)
;3223  ; IT WILL BE CAUGHT HERE AND THE NEXT 15 TESTS SHOULD BE RUN
;3224  ; TO REPORT WHAT DEVICES ARE AT WHAT TR LEVELS.
;3225  ;
;3226  ; DATA DESCRIPTION:
;3227  ; **SUBTESTS 1-4**
;3228  ;
;3229  ; EXPECTED SILO CONTENTS FOR COMMAND ADDRESS (C/A)
;3230  ; EXPECTED SILO CONTENTS FOR SECONDARY PATTERN (SEE NOTE)
;3231  ;  (THE NEXT 11 DATA WORDS ARE 11 LOCATIONS OF SILO)
;3232  ; ARB (SHOULD BE ZERO ELSE SOMEONE'S TR IS RAISED)
;3233  ; C/A (SEE BELOW FOR FUNCTION)
;3234  ; SECONDARY PATTERN
;3235  ; NO RESPONSE (ZERO)
;3236  ; ARB
;3237  ; C/A
;3238  ; SECONDARY PATTERN
;3239  ; NO RESPONSE
;3240  ; ARB
;3241  ; C/A
;3242  ; SECONDARY PATTERN
;3243  ; ID[FAULT] (ID ADRS:1B)
;3244  ; ID[SBI.ERR] (ID ADRS:19)
;3245  ;
;3246  ;
;3247  ; **SUBTEST 5** (ONLY ONE RETRY)
;3248  ;
;3249  ; EXPECTED SILO CONTENTS FOR COMMAND ADDRESS (READ.INT.SUM)
;3250  ; EXPECTED SILO CONTENTS FOR SECONDARY DATA PATTERN (HOLD)
;3251  ; ARB
;3252  ; C/A
;3253  ; HOLD
;3254  ; 9 WORDS OF ZERO
;3255  ; ID[FAULT] (ID ADRS:1B)
;3256  ; ID[SBI.ERR] (ID ADRS:19)
;3257  ;
;3258  ;  FUNCTIONS:
;3259  ;    SUBTEST 1 - READ.P
;3260  ;    SUBTEST 2 - WRITE.P
;3261  ;    SUBTEST 3 - LOCKREAD.P
;3262  ;    SUBTEST 4 - LOCKWRITE.P
```

```
            ;3263 ;   SUBTEST 5 - READ.INT.SUM
            ;3264 ;
            ;3265 ;  NOTE: SECONDARY PATTERNS ARE GENERATED ON:
            ;3266 ;   WRITE REFERENCES - WRITE DATA FOLLOWS
            ;3267 ;   INTERLOCK READ REFERENCES - SBI INTLK ASSERTED
            ;3268 ;   INTERRUPT SUMMARY READS - HOLD ASSERTED NEXT CYCLE
            ;3269 ;--
            ;3270 ;*****************************************************************
            ;3271 =0
U 1152. 0000.003D.0180.0800.0000.1216  ;3272 CATST: NEWTST
U 1153. 0018.0038.6180.09F8.0000.1156  ;3273  RC[0F]_K[.F]
            ;3274 ;/////////////////////////////////////////////////////////////////
            ;3275 ;*
            ;3276 ; SUBTEST 1 VA=20000000, READ.P
            ;3277 ;-
U 1156. 0000.003D.0180.0800.0000.122D  ;3278 =0 SUBTEST
U 1157. 0000.003C.0180.0800.0000.1158  ;3279  NOP
U 1158. 0000.003D.0180.0800.0000.1228  ;3280 =0 ERLOOP
U 1159. 0000.003C.0180.0800.0000.115A  ;3281  NOP
U 115A. 0000.003D.0180.0800.0000.1250  ;3282 =0 CALL,J/CLCPTO  ; CLEAR SBI.ERR
U 115B. 0000.003C.0180.0800.0000.115C  ;3283  NOP
U 115C. 0000.003D.0180.0800.0000.1266  ;3284 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 115D. 0000.003C.0180.0800.0000.115E  ;3285  NOP
U 115E. 0000.003D.0180.0800.0000.1092  ;3286 =0 CALL,J/CPUCA  ; GENERATE EXPECTED DATA
U 115F. 0000.003C.2180.0800.0084.72FE  ;3287  SC_K[.14]
U 12FE. 0000.003C.0980.0800.0084.B2FF  ;3288  SC_SC-K[.2]
U 12FF. 0801.001C.0180.0800.0000.1300  ;3289  D_D.ORNOT.MASK
U 1300. 0001.003C.0180.0A88.0000.1301  ;3290  R[1]_D   ; SAVE EXP C/A
U 1301. 0003.003C.0180.0AF0.0000.1160  ;3291  R[0E]_0   ; SAVE EXP. SECONDARY
U 1160. 0000.003D.0180.0800.0000.1280  ;3292 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 1161. 0000.003C.8980.0800.0084.7302  ;3293  SC_K[.D]
U 1302. 0003.001C.0180.0AF8.0000.1162  ;3294  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 1162. 0000.003D.0180.0800.0000.1010  ;3295 =0 CALL,J/TRO  ; D_NON-EXISTENT I/O ADRS
U 1163. 0001.003C.7180.3C00.0200.1164  ;3296  VA_D,ID[COMP]_D
U 1164. 0000.003D.0180.C800.0000.10A0  ;3297 =0 CALL,J/SLOLCK,MCT/READ.P ; PUTS SILO IN RC'S
U 1165. 0800.003C.1980.0A08.1404.7040  ;3298  STATE_K[ZERO],D_R[1] ; TELLS SILRTN SECONDARY IS NOT EXP
U 1040. 0018.0039.0D80.0AE8.0000.128C  ;3299 =00 CALL,J/SILRTN,R[0D]_K[.3] ; EXAMINES RC'S FOR SILO DATA
U 1041. 0000.003C.0180.0800.0000.1043  ;3300  J/ER1   ; RETURN 1=> ERROR OCCURRED
U 1042. 0000.003C.0180.0800.0000.1167  ;3301  J/ESUB1   ; RETURN2 => OK
U 1043. 0001.003C.0180.09F0.0000.1303  ;3302 ER1: RC[0E]_D  ; PUT EXP DATA FOR TYPE OUT
U 1303. 0800.003C.0180.0A70.0000.1166  ;3303  D_R[0E]   ;....
U 1166. 0001.003D.0180.09E8.0000.122B  ;3304 =0 ERROR2,RC[0D]_D  ; C/A INCORRECT OR SBI CORRUPTED
U 1167. 0000.003C.0180.0800.0000.1168  ;3305 ESUB1: NOP
            ;3306 ;/////////////////////////////////////////////////////////////////
            ;3307 ;*
            ;3308 ; SUBTEST 2 VA=20000000, WRITE.P
            ;3309 ;-
U 1168. 0000.003D.0180.0800.0000.122D  ;3310 =0 SUBTEST
U 1169. 0000.003C.0180.0800.0000.116A  ;3311  NOP
U 116A. 0000.003D.0180.0800.0000.1228  ;3312 =0 ERLOOP
U 116B. 0000.003C.0180.0800.0000.116C  ;3313  NOP
U 116C. 0000.003D.0180.0800.0000.1250  ;3314 =0 CALL,J/CLCPTO  ; CLEAR SBI.ERR
U 116D. 0000.003C.0180.0800.0000.116E  ;3315  NOP
U 116E. 0000.003D.0180.0800.0000.1266  ;3316 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 116F. 0000.003C.0180.0800.0000.1170  ;3317  NOP
```

```
U 1170. 0000.003D.0180.0800.0000.109C  ;3318 =0 CALL,J/WRITCA  ; GENERATE EXPECTED DATA
U 1171. 0001.003C.0180.0A88.0000.1172  ;3319  R[1]_D   ; SAVE EXP C/A
U 1172. 0000.003D.0180.0800.0000.108E  ;3320 =0 CALL,J/WDPAT
U 1173. 0001.003C.0180.0AF0.0000.1174  ;3321  R[0E]_D   ; SAVE EXP. SECONDARY
U 1174. 0000.003D.0180.0800.0000.1280  ;3322 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 1175. 0000.003C.8980.0800.0084.7304  ;3323  SC_K[.D]
U 1304. 0003.001C.0180.0AF8.0000.1176  ;3324  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 1176. 0000.003D.0180.0800.0000.1010  ;3325 =0 CALL,J/TR0  ; D_NON-EXISTENT I/O ADRS
U 1177. 0001.003C.7180.3C00.0200.1178  ;3326  VA_D,ID[COMP]_D
U 1178. 0000.003D.0180.A800.0000.10A0  ;3327 =0 CALL,J/SLOLCK,MCT/WRITE.P ; PUTS SILO IN RC'S
U 1179. 0800.003C.0580.0A08.1404.7048  ;3328  STATE_K[.1],D_R[1] ; TELLS SILRTN SECONDARY IS EXP
U 1048. 0018.0039.0D80.0AE8.0000.128C  ;3329 =00 CALL,J/SILRTN,R[0D]_K[.3] ; EXAMINES RC'S FOR SILO DATA
U 1049. 0000.003C.0180.0800.0000.104B  ;3330  J/ER2   ; RETURN 1=> ERROR OCCURRED
U 104A. 0000.003C.0180.0800.0000.117B  ;3331  J/ESUB2   ; RETURN2 => OK
U 104B. 0001.003C.0180.09F0.0000.1305  ;3332 ER2: RC[0E]_D  ; PUT EXP DATA FOR TYPE OUT
U 1305. 0800.003C.0180.0A70.0000.117A  ;3333  D_R[0E]   ;...
U 117A. 0001.003D.0180.09E8.0000.122B  ;3334 =0 ERROR2,RC[0D]_D  ; C/A INCORRECT OR SBI CORRUPTED
U 117B. 0000.003C.0180.0800.0000.117C  ;3335 ESUB2: NOP
        ;3336 ;////////////////////////////////////////////////////////////
        ;3337 ;*
        ;3338 ; SUBTEST 3 VA=20000000, LOCKREAD.P
        ;3339 ;-
U 117C. 0000.003D.0180.0800.0000.122D  ;3340 =0 SUBTEST
U 117D. 0000.003C.0180.0800.0000.117E  ;3341  NOP
U 117E. 0000.003D.0180.0800.0000.1228  ;3342 =0 ERLOOP
U 117F. 0000.003C.0180.0800.0000.1180  ;3343  NOP
U 1180. 0000.003D.0180.0800.0000.1250  ;3344 =0 CALL,J/CLCPTO  ; CLEAR SBI.ERR
U 1181. 0000.003C.0180.0800.0000.1182  ;3345  NOP
U 1182. 0000.003D.0180.0800.0000.1266  ;3346 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 1183. 0000.003C.0180.0800.0000.1184  ;3347  NOP
U 1184. 0000.003D.0180.0800.0000.1092  ;3348 =0 CALL,J/CPUCA  ; GENERATE EXPECTED DATA
U 1185. 0000.003C.2180.0800.0084.7306  ;3349  SC_K[.14]
U 1306. 0801.001C.0180.0800.0000.1307  ;3350  D_D.ORNOT.MASK
U 1307. 0001.003C.0180.0A88.0000.1186  ;3351  R[1]_D   ; SAVE EXP C/A
U 1186. 0000.003D.0180.0800.0000.1010  ;3352 =0 CALL,J/TR0
U 1187. 0500.003C.0180.0800.0000.1308  ;3353  D_D.LEFT
U 1308. 0001.003C.0180.0AF0.0000.1188  ;3354  R[0E]_D   ; SAVE EXP. SECONDARY
U 1188. 0000.003D.0180.0800.0000.1280  ;3355 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 1189. 0000.003C.8980.0800.0084.7309  ;3356  SC_K[.D]
U 1309. 0003.001C.0180.0AF8.0000.118A  ;3357  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 118A. 0000.003D.0180.0800.0000.1010  ;3358 =0 CALL,J/TR0  ; D_NON-EXISTENT I/O ADRS
U 118B. 0001.003C.7180.3C00.0200.118C  ;3359  VA_D,ID[COMP]_D
U 118C. 0000.003D.0180.E800.0000.10A0  ;3360 =0 CALL,J/SLOLCK,MCT/LOCKREAD.P ; PUTS SILO IN RC'S
U 118D. 0800.003C.0580.0A08.1404.7050  ;3361  STATE_K[.1],D_R[1] ; TELLS SILRTN SECONDARY IS EXP
U 1050. 0018.0039.0D80.0AE8.0000.128C  ;3362 =00 CALL,J/SILRTN,R[0D]_K[.3] ; EXAMINES RC'S FOR SILO DATA
U 1051. 0000.003C.0180.0800.0000.1053  ;3363  J/ER3   ; RETURN 1=> ERROR OCCURRED
U 1052. 0000.003C.0180.0800.0000.118F  ;3364  J/ESUB3   ; RETURN2 => OK
U 1053. 0001.003C.0180.09F0.0000.130A  ;3365 ER3: RC[0E]_D  ; PUT EXP DATA FOR TYPE OUT
U 130A. 0800.003C.0180.0A70.0000.118E  ;3366  D_R[0E]   ;...
U 118E. 0001.003D.0180.09E8.0000.122B  ;3367 =0 ERROR2,RC[0D]_D  ; C/A INCORRECT OR SBI CORRUPTED
U 118F. 0000.003C.0180.0800.0000.1190  ;3368 ESUB3: NOP
        ;3369 ;////////////////////////////////////////////////////////////
        ;3370 ;*
        ;3371 ; SUBTEST 4 VA=20000000, LOCKWRITE.P
        ;3372 ;-
```

```
U 1190. 0000.003D.0180.0800.0000.122D  ;3373 =0 SUBTEST
U 1191. 0000.003C.0180.0800.0000.1192  ;3374  NOP
U 1192. 0000.003D.0180.0800.0000.1228  ;3375 =0 ERLOOP
U 1193. 0000.003C.0180.0800.0000.1194  ;3376  NOP
U 1194. 0000.003D.0180.0800.0000.1250  ;3377 =0 CALL,J/CLCPTO  ; CLEAR SBI.ERR
U 1195. 0000.003C.0180.0800.0000.1196  ;3378  NOP
U 1196. 0000.003D.0180.0800.0000.1266  ;3379 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 1197. 0000.003C.0180.0800.0000.1198  ;3380  NOP
U 1198. 0000.003D.0180.0800.0000.108E  ;3381 =0 CALL,J/WDPAT  ; GENERATE EXPECTED DATA
U 1199. 0001.003C.7D80.0AF0.0084.730B  ;3382  R[0E]_D,SC_K[.18] ; SAVE EXP. SECONDARY
U 130B. 0801.0034.0180.0800.0000.130C  ;3383  D_D.AND.MASK
U 130C. 0000.003C.0580.0800.0084.B30D  ;3384  SC_SC-K[.1]
U 130D. 0801.001C.0180.0800.0000.130E  ;3385  D_D.ORNOT.MASK
U 130E. 0000.003C.0980.0800.0084.B30F  ;3386  SC_SC-K[.2]
U 130F. 0801.0034.0180.0800.0000.1310  ;3387  D_D.AND.MASK
U 1310. 0819.0030.0580.0800.0000.1311  ;3388  D_D.OR.K[.1]
U 1311. 0001.003C.0180.0A88.0000.119A  ;3389  R[1]_D  ; SAVE EXP C/A
U 119A. 0000.003D.0180.0800.0000.1280  ;3390 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 119B. 0000.003C.8980.0800.0084.7312  ;3391  SC_K[.D]
U 1312. 0003.001C.0180.0AF8.0000.119C  ;3392  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 119C. 0000.003D.0180.0800.0000.1010  ;3393 =0 CALL,J/TRO  ; D_NON-EXISTENT I/O ADRS
U 119D. 0001.003C.7180.3C00.0200.119E  ;3394  VA_D,ID[COMP]_D
U 119E. 0000.003D.0180.B800.0000.10A0  ;3395 =0 CALL,J/SLOLCK,MCT/LOCKWRITE.P ; PUTS SILO IN RC'S
U 119F. 0800.003C.0580.0A08.1404.7054  ;3396  STATE_K[.1],D_R[1] ; TELLS SILRTN SECONDARY IS EXP
U 1054. 0018.0039.0D80.0AE8.0000.128C  ;3397 =00 CALL,J/SILRTN,R[0D]_K[.3] ; EXAMINES RC'S FOR SILO DATA
U 1055. 0000.003C.0180.0800.0000.1057  ;3398  J/ER4  ; RETURN 1=> ERROR OCCURRED
U 1056. 0000.003C.0180.0800.0000.11A1  ;3399  J/ESUB4  ; RETURN2 => OK
U 1057. 0001.003C.0180.09F0.0000.1313  ;3400 ER4: RC[0E]_D  ; PUT EXP DATA FOR TYPE OUT
U 1313. 0800.003C.0180.0A70.0000.11A0  ;3401  D_R[0E]  ; ...
U 11A0. 0001.003D.0180.09E8.0000.122B  ;3402 =0 ERROR2,RC[0D]_D  ; C/A INCORRECT OR SBI CORRUPTED
U 11A1. 0000.003C.0180.0800.0000.11A2  ;3403 ESUB4: NOP
            ;3404 ;////////////////////////////////////////////////////////////
            ;3405 ;*
            ;3406 ; SUBTEST 5 VA=20000000, READ.INT.SUM
            ;3407 ;-
U 11A2. 0000.003D.0180.0800.0000.122D  ;3408 =0 SUBTEST
U 11A3. 0000.003C.0180.0800.0000.11A4  ;3409  NOP
U 11A4. 0000.003D.0180.0800.0000.1228  ;3410 =0 ERLOOP
U 11A5. 0000.003C.0180.0800.0000.11A6  ;3411  NOP
U 11A6. 0000.003D.0180.0800.0000.1250  ;3412 =0 CALL,J/CLCPTO  ; CLEAR SBI.ERR
U 11A7. 0000.003C.0180.0800.0000.11A8  ;3413  NOP
U 11A8. 0000.003D.0180.0800.0000.1266  ;3414 =0 CALL,J/HLD.UNJ  ; HOLD FOR 16,UNJAM FOR 16,HOLD FOR 16
U 11A9. 0000.003C.0180.0800.0000.11AA  ;3415  NOP
U 11AA. 0000.003D.0180.0800.0000.108E  ;3416 =0 CALL,J/WDPAT  ; GENERATE EXPECTED DATA
U 11AB. 0000.003C.2180.0800.0084.7314  ;3417  SC_K[.14]
U 1314. 0000.003C.09F8.0800.0084.B315  ;3418  SC_SC-K[.2],Q_0
U 1315. 0001.201C.01C0.0800.0000.1316  ;3419  Q_Q.ORNOT.MASK
U 1316. 081D.0014.0180.0800.0000.1317  ;3420  ALU_D+Q,D_ALU
U 1317. 0819.0030.0580.0800.0000.1318  ;3421  D_D.OR.K[.1]
U 1318. 0001.003C.0180.0A88.0000.1319  ;3422  R[1]_D  ; SAVE EXP C/A
U 1319. 0018.0038.0580.0AF0.0000.11AC  ;3423  R[0E]_K[.1]  ; SAVE EXP. SECONDARY
U 11AC. 0000.003D.0180.0800.0000.1280  ;3424 =0 CALL,J/SILCLR  ; CLEAR OUT SILO
U 11AD. 0000.003C.8980.0800.0084.731A  ;3425  SC_K[.D]
U 131A. 0003.001C.0180.0AF8.0000.11AE  ;3426  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 11AE. 0000.003D.0180.0800.0000.1010  ;3427 =0 CALL,J/TRO  ; D_NON-EXISTENT I/O ADRS
```

```
U 11AF. 0001.003C.7180.3C00.0200.11B0   ;3428  VA_D,ID[COMP]_D
U 11B0. 0000.003D.0180.D800.0000.10A0   ;3429 =0 CALL,J/SLOLCK,MCT/READ.INT.SUM ; PUTS SILO IN RC'S
U 11B1. 0800.003C.0580.0A08.1404.7080   ;3430  STATE_K[.1],D_R[1] ; TELLS SILRTN SECONDARY IS EXP
U 1080. 0018.0039.0580.0AE8.0000.128C   ;3431 =00 CALL,J/SILRTN,R[0D]_K[.1] ; EXAMINES RC'S FOR SILO DATA
U 1081. 0000.003C.0180.0800.0000.1083   ;3432  J/ER5    ; RETURN 1=> ERROR OCCURRED
U 1082. 0000.003C.0180.0800.0000.11B3   ;3433  J/ESUB5    ; RETURN2 => OK
U 1083. 0001.003C.0180.09F0.0000.131B   ;3434 ER5: RC[0E]_D   ; PUT EXP DATA FOR TYPE OUT
U 131B. 0800.003C.0180.0A70.0000.11B2   ;3435  D_R[0E]    ; ...
U 11B2. 0001.003D.0180.09E8.0000.122B   ;3436 =0 ERROR2,RC[0D]_D   ; C/A INCORRECT OR SBI CORRUPTED
U 11B3. 0000.003C.0180.0800.0000.11B4   ;3437 ESUB5: NOP
```

```
;3438  .PAGE "TEST 17F SBI NEXUS RESPONSE TEST"
;3439 ;*****************************************************************
;3440 ;+*
;3441 ; TEST 17F SBI NEXUS RESPONSE TEST
;3442 ;
;3443 ; TEST DESCRIPTION:
;3444 ;
;3445 ;       THIS TEST LOOPS THROUGH TR LEVELS 0-F LOOKING FOR
;3446 ; A RESPONDING DEVICE (READ REFERENCES THAT DO NOT TIME OUT).
;3447 ; ONCE A DEVICE HAS BEEN FOUND, THE TEST DOES THE FOLLOWING:
;3448 ;  A. TYPE THE NAME OF THE DEVICE AND IT'S TR LEVEL
;3449 ;     TO THE CONSOLE.
;3450 ;     IF THE ADAPTER CODE BITS IN THE CONFIGURATION
;3451 ;     REGISTER ARE INCORRECT, "UNKNOWN DEVICE AT TR XX"
;3452 ;     WILL BE TYPED.
;3453 ;
;3454 ;  B. CHECK TO SEE IF AN INTERRUPT WAS ASSERTED ON THE
;3455 ;     REFERENCE.
;3456 ;     FLAGGED AS ERROR CODE 0.
;3457 ;
;3458 ;
;3459 ;  E. CHECK TO SEE THAT THE EEVICE TRANSMITTED THE PROPER
;3460 ;     TAG AND MASK BITS FOR THE READ DATA RESPONSE
;3461 ;   TAG<2:0>=0, MASK<3:0>=0
;3462 ;     FLAGGED AS ERROR CODE 1.
;3463 ;     MUST HAVE TRANSMITTED CPU ID, OR CPU WOULD HAVE
;3464 ;     TIMED OUT.
;3465 ;
;3466 ;  E. CHECK TO SEE THAT THE DEVICE ARBITRATED AT THE
;3467 ;     PROPER TR LEVEL AT THE PROPER TIME.
;3468 ;   READ DATA - 1, SILO <15:00>= 2**TR
;3469 ;     FLAGGED AS ERROR CODE 2.
;3470 ;
;3471 ;
;3472 ;  *. NOTE: IN ORDER TO GUARUNTEE THAT THE READ DATA
;3473 ;   RESPONSE APPEARS IN THE SILO, THE SILO IS LOCKED
;3474 ;   ON THE FIRST OCCURRENCE OF READ DATA TAG
;3475 ;   TO THE CPU.  GIVEN THAT A DEVICE MAY RESPOND
;3476 ;   WITH READ DATA UP TO 512 CYCLES FROM COMMAND
;3477 ;   ADDRESS, THE CYCLES CONTAINING COMMAND ADDRESS
;3478 ;   AND ACK MAY OR MAY NOT APPEAR IN THE ERROR DATA
;3479 ;
;3480 ; LOGIC DESCRIPTION:
;3481 ;
;3482 ; LOGIC ON SBL, SBH, TRS, AND THE RESPONDING DEVICE ARE ALL
;3483 ; CHECKED HERE.
;3484 ; KEEP IN MIND THAT ALL THE BITS IN THE SILO HAVE
;3485 ; NOT BEEN CHECKED IN THE ASSERTED ("1") STATE.
;3486 ;
;3487 ;
;3488 ; DATA DESCRIPTION:
;3489 ;
;3490 ; CONFIGURATION REGISTER ADDRESS OF RESPONDING DEVICE
;3491 ; CONTENTS OF THE CONFIGURATION REGISTER
;3492 ; ERROR CODE: (SEE TEST DESCRIPTION)
```

```
            ;3493 ; ID[VECTOR] ID ADRS:0D
            ;3494 ; ID[SBI.ERR] ID ADRS:19
            ;3495 ; ID[FAULT] ID ADRS:1B
            ;3496 ;*****THE REST OF THE ERROR DATA IS FROM THE SILO****
            ;3497 ;*****THE LAST TWO DATA WORDS SHOULD BE "ARB" AND "READ DATA"*****
            ;3498 ;*****THE REST OF THE PROTOCOL MAY OR MAY NOT APPEAR*****
            ;3499 ; ...
            ;3500 ; ...
            ;3501 ; ...
            ;3502 ; ...
            ;3503 ; ...
            ;3504 ; ...
            ;3505 ; ...
            ;3506 ; DEVICE ARBITRATION
            ;3507 ; READ DATA
            ;3508 ;
            ;3509 ;
            ;3510 ;--
            ;3511 ;*************************************************************************
            ;3512 =0
U 11B4. 0000.003D.0180.0800.0000.1216  ;3513 TRTST: NEWTST
U 11B5. 0F18.003B.6180.09F8.0000.131C  ;3514  RC[0F]_K[.F],D_0
U 131C. 0000.003C.9180.3C00.0000.11B6  ;3515  ID[24]_D  ; INIT TR LEVEL AT 0
U 11B6. 0000.003D.0180.0800.0000.10F8  ;3516 =0 CALL[MNTID]  ; GET CPU'S ID
U 11B7. 0000.003C.7580.3C00.0000.11B8  ;3517  ID[MAINT]_D  ; LEAVE IN ID[MAINT] FOR WHOLE TEST
U 11B8. 0000.003D.0180.0800.0000.1228  ;3518 =0 ERLOOP
U 11B9. 0000.003C.0180.0800.0000.11BA  ;3519  NOP
            ;3520 =0
U 11BA. 0000.003D.0180.0800.0000.1250  ;3521 GETDEV: CALL[CLCPTO]  ; CLEAR TIME OUT BIT
U 11BB. 0000.003C.0180.0800.0000.11BC  ;3522  NOP
U 11BC. 0000.003D.0180.0800.0000.1266  ;3523 =0 CALL[HLD.UNJ]  ; DO AN UNJAM SEQUENCE
U 11BD. 0000.003C.0180.0800.0000.11BE  ;3524  NOP
U 11BE. 0000.003D.0180.0800.0000.1253  ;3525 =0 CALL[ENFLTI]  ; ENABLE FAULT INTERRUPTS
U 11BF. 0000.003C.0180.0800.0000.11C0  ;3526  NOP
U 11C0. 0000.003D.0180.0800.0000.1280  ;3527 =0 CALL[SILCLR]  ; CLEAR OUT SBI SILO
U 11C1. 0000.003C.91F0.2C00.0000.131D  ;3528  Q_ID[24]
U 131D. 0C01.203C.0180.09E8.0000.11C2  ;3529  D_Q,RC[0D]_Q  ; "CONFIG" NEEDS TR IN RC[0D]
U 11C2. 0000.193D.0180.0800.0000.1010  ;3530 =0 CALL[TR0],D3-0?  ; GET ADDRESS FOR THIS TR
U 11C3. 0001.003C.0180.0800.0200.11C4  ;3531  VA_ALU,ALU_D
U 11C4. 0000.003D.0180.0800.0000.12C9  ;3532 =0 CALL[LK1RDT]  ; PATTERN LOCKS SILO ON FIRST READ DATA
U 11C5. 0000.003C.8980.0800.0084.731E  ;3533  SC_K[.D]  ; PREPARE FOR TIME OUT TRAP
U 131E. 0000.003C.7180.3C00.0000.131F  ;3534  ID[COMP]_D  ; LOCK THE SILO ON READ DATA
            ;3535
U 131F. 0003.001C.0180.CAF8.0000.1320  ;3536  MCT/READ.P,R[0F]_NOT.MASK
            ;3537
U 1320. 0000.003C.9580.3E78.0010.1321  ;3538  ALU_R[0F],CLK.UBCC,ID[25]_D ; TIME OUT? (SAVE READ DATA)
U 1321. 0000.013C.0180.0800.0000.11C6  ;3539  Z?
U 11C6. 0000.003C.0180.0800.0000.11C8  ;3540 =0 J/TYPMSG  ; RESPONSE, TYPE DEVICE NAME
U 11C7. 0000.003C.0180.0800.0000.1337  ;3541  J/NEXTTR  ; NO RESPONSE, GET NEXT TR
            ;3542
            ;3543 =0
U 11C8. 0000.003D.0180.0800.0000.12BD  ;3544 TYPMSG: CALL[DECFLG]  ; DECREMENT THE TYPE OUT FLAG
U 11C9. 0000.003C.95F0.2C00.0000.1322  ;3545  Q_ID[25]  ; "CONFIG" NEEDS READ DATA IN D
U 1322. 0C00.003C.0180.0800.0000.11CA  ;3546  D_Q
U 11CA. 0000.003D.0180.0800.0000.129C  ;3547 =0 CALL[CONFIG]  ; TYPE "XXXXXX AT TR XX"
```

```
U 11CB. 0000,003C,0180,0800,0000,11CC  ;3548  NOP
        ;3549
U 11CC. 0000,003D,0180,0800,0000,10A0  ;3550 =0 CALL[SLOLCK]  ; PUT SILO IN RC[0E-0]
        ;3551
U 11CD. 0F00,003C,0180,0800,0000,1323  ;3552  D_0   ; STROBE THE INTERRUPT
U 1323. 0000,003C,3D80,3C00,0000,1324  ;3553  ID[PSL]_D  ; SET LOWEST IPL ACTIVE
U 1324. 0000,003C,0180,0800,4000,1325  ;3554  IEK/ISTR
U 1325. 0000,003C,0180,0800,0000,1326  ;3555  NOP   ; WAIT
U 1326. 0000,003C,35F0,2C00,0000,1327  ;3556  Q_ID[VECTOR]
U 1327. 0019,2034,8180,0800,0010,1328  ;3557  ALU_Q.AND.K[.3FF],CLK.UBCC ; MASK FOR VECTOR ONLY
U 1328. 0000,013C,0180,0800,0000,11CE  ;3558  Z?
U 11CE. 0000,003C,0180,0800,0000,11D0  ;3559 =0 J/ERR00   ; AN INTERRUPT OCCURRED, MAKE CALL
U 11CF. 0000,003C,0180,0800,0000,1329  ;3560  J/FNDRDT  ; NO INTERRUPT, START CHECKING SILO
        ;3561
        ;3562 =0
U 11D0. 0000,003D,0180,0800,0000,12C0  ;3563 ERR00: CALL[SBIDAT]  ; GET THE REST OF THE ERROR DATA
U 11D1. 0000,003C,0180,0800,0000,11D2  ;3564  NOP
U 11D2. 0003,003D,0180,09E0,0000,122B  ;3565 =0 ERROR2,RC[0C]_0  ; AN INTERRUPT IS ASSERTED ON THE READ
U 11D3. 0000,003C,0180,0800,0000,1337  ;3566  J/NEXTTR  ; TO THE CONFIGURATION REGISTER
        ;3567
        ;3568
        ;3569
U 1329. 0000,003C,1980,0800,0084,732A  ;3570 FNDRDT: SC_K[ZERO] ; START LOOKING IN RC'S FOR THE READ DATA
U 132A. 0010,0038,0180,0830,0010,132B  ;3571 FNDLP: ALU_RC(SC),CLK.UBCC
U 132B. 0000,013C,0180,0800,0000,11D4  ;3572  Z?
U 11D4. 0000,003C,0180,0800,0100,132C  ;3573 =0 J/TAGMSK,FE_SC  ; FOUND SOMETHING, CHECK IT.
U 11D5. 0000,003C,0180,0800,0080,D32A  ;3574  J/FNDLP,SC_SC+1
U 132C. 0818,0038,5980,0800,0000,132D  ;3575 TAGMSK: D_K[.7F]
U 132D. 0000,003C,6580,0800,0084,732E  ;3576  SC_K[.10]  ; TAG/MASK SHOULD = 0
U 132E. 0000,003C,09F8,0800,0084,932F  ;3577  SC_SC+K[.2],Q_0
U 132F. 0D00,003C,0180,0800,0081,1330  ;3578  D_DAL.SC,SC_FE
U 1330. 0011,0034,0180,0830,0010,1331  ;3579  ALU_D[AND]RC(SC),CLK.UBCC
U 1331. 0000,013C,0580,0800,0104,91D6  ;3580  Z?,FE_SC+K[.1]  ; SET UP FOR ARB IN THE PREV.CYCLE
U 11D6. 0000,003C,0180,0800,0000,11D8  ;3581 =0 J/ERR01   ; NO GOOD, MAKE CALL
U 11D7. 0000,003C,0180,0800,0000,1332  ;3582  J/DEVARB
        ;3583
        ;3584 =0
U 11D8. 0000,003D,0180,0800,0000,12C0  ;3585 ERR01: CALL[SBIDAT]  ; GET THE REST OF THE ERROR DATA
U 11D9. 0000,003C,0180,0800,0000,11DA  ;3586  NOP
U 11DA. 0018,0039,0580,09E0,0000,122B  ;3587 =0 ERROR2,RC[0C]_K[.1] ; READ DATA TAG AND MASK ARE NOT
U 11DB. 0000,003C,0180,0800,0000,1337  ;3588  J/NEXTTR  ; CORRECT (=0).
        ;3589
        ;3590
U 1332. 0000,003C,91F0,2C00,0000,1333  ;3591 DEVARB: Q_ID[24]  ; SEE IF THIS DEVICE ARB'D AT THE
U 1333. 0019,2034,6180,0800,0082,1334  ;3592  SC_ALU,ALU_Q.AND.K[.F] ; RIGHT TR LEVEL
U 1334. 0803,001C,0180,0800,0081,1335  ;3593  D_NOT.MASK,SC_FE
U 1335. 0011,0034,0180,0830,0010,1336  ;3594  ALU_D[AND]RC(SC),CLK.UBCC ; CAN'T XOR<-ARB/ACK CAN COME TOGETHER
U 1336. 0000,013C,0180,0800,0000,11DC  ;3595  Z?
U 11DC. 0000,003C,0180,0800,0000,1337  ;3596 =0 J/NEXTTR
U 11DD. 0000,003C,0180,0800,0000,11DE  ;3597  J/ERR02   ; NO GOOD, MAKE CALL
        ;3598
        ;3599 =0
U 11DE. 0000,003D,0180,0800,0000,12C0  ;3600 ERR02: CALL[SBIDAT]  ; GET THE REST OF THE ERROR DATA
U 11DF. 0000,003C,0180,0800,0000,11E0  ;3601  NOP
U 11E0. 0018,0039,0980,09E0,0000,122B  ;3602 =0 ERROR2,RC[0C]_K[.2] ; DEVICE IS NOT ARBITRATING AT
```

```
U 11E1, 0000,003C,0180,0800,0000,1337  ;3603  J/NEXTTR  ; THE PROPER TR (AT THE RIGHT TIME)
          ;3604
U 1337, 0000,003C,91F0,2C00,0000,1338  ;3605 NEXTTR: Q_ID[24]  ; GET THE TR LEVEL
U 1338, 001B,A020,6180,0800,0010,1339  ;3606  ALU_Q.0XT[BYTE].XOR.K[.F],CLK.UBCC ; UP TO F YET?
U 1339, 0819,2114,0580,0800,0000,11E2  ;3607  Z?,D_Q+K[.1]
U 11E2, 0000,003C,9180,3C00,0000,11BA  ;3608 =0 J/GETDEV,ID[24]_D ; REPEAT WITH NEXT TR
U 11E3, 0000,003C,0180,0800,0000,11E4  ;3609  NOP
          ;3610
```

```
            ;3611  .PAGE "TEST 180 AT LEAST ONE MEMORY CONTROLLER PRESENT TEST"
            ;3612 ;*****************************************************************
            ;3613 ;++
            ;3614 ; TEST 180 AT LEAST ONE MEMORY CONTROLLER PRESENT TEST
            ;3615 ;
            ;3616 ; TEST DESCRIPTION:
            ;3617 ;
            ;3618 ;      THIS TEST SEARCHES ALL TR LEVELS LOOKING FOR A MEMORY
            ;3619 ; CONTROLLER, IF ONE IS FOUND, THE TEST ENDS.
            ;3620 ; IF NO MEMORY CONTROLLERS ARE FOUND, THE MESSAGES:
            ;3621 ; "?NO MS780-E/H MEMORY" AND "DEPOSIT MS780-E/H I/O BASE ADDR
            ;3622 ; IN RC[0] & TYPE LO" ARE TYPED TO THE CONSOLE, AND THE
            ;3623 ; TEST LOOPS ON REFERENCES TO THAT ADDRESS.
            ;3624 ;
            ;3625 ; LOGIC DESCRIPTION:
            ;3626 ;
            ;3627 ; LOGIC ON MSB (M8214) IS TESTED HERE
            ;3628 ;
            ;3629 ; DATA DESCRIPTION:
            ;3630 ; NONE
            ;3631 ;
            ;3632 ;--
            ;3633 ;*****************************************************************
            ;3634 ;/////////////////////////////////////////////////////////////////
            ;3635 =0
U 11E4, 0018.0039.1980.09F8.0000.1216  ;3636 ANYMEM: NEWTST[ZERO]
U 11E5, 0003.003C.0180.0AF0.0000.133A  ;3637  R[0E]_0   ; SET NO MA780 FLAG
U 133A, 0000.003C.8980.0800.0084.733B  ;3638 START: SC_K[.D]  ; SET UP FOR TIME OUT
U 133B, 0003.001C.0180.0AF8.0000.11E6  ;3639  R[0F]_NOT.MASK  ; SET TIME OUT BIT FOR TRAP CATCHER
U 11E6, 0000.193D.0180.0800.0000.1010  ;3640 =0 CALL,J/TR0,D3-0? ; GET ADDRESS FOR THIS TR
U 11E7, 0001.203C.0180.09E8.0000.133C  ;3641  RC[0D]_Q  ; Q GOT TR LEVEL, SAVE IT
U 133C, 0001.003C.0180.0800.0200.133D  ;3642  VA_D   ; D GOT ADDRESS FOR THIS TR
U 133D, 0000.003C.0180.C800.0000.133E  ;3643  MCT/READ.P  ; DO THE REFERENCE
U 133E, 0000.003C.0180.0A78.0010.133F  ;3644  ALU_R[0F],CLK.UBCC ; TIME OUT OCCUR?
U 133F, 0000.013C.0180.0800.0000.11E8  ;3645  Z?
U 11E8, 0000.003C.0180.0800.0000.11EA  ;3646 =0 J/DEV   ; NO, SEE IF MEMORY
U 11E9, 0810.0038.0180.0968.0000.1341  ;3647  J/DONE,D_RC[0D]  ; YES, SEE IF LAST TR
            ;3648 =0
U 11EA, 0200.003C.0180.0800.0000.11EB  ;3649 DEV: D_D.RIGHT2  ; PUT ADAPTER CODE BITS<6:3> INTO
U 11EB, 0600.003C.0180.0800.0000.1340  ;3650  D_D.RIGHT  ; D REGISTER BITS<3:0>
U 1340, 0000.193C.0180.0800.0000.1070  ;3651  D3-0?   ;  1 FOR 4K CTRLR, 2 FOR 16K CTRLR
            ;3652      ; 8 FOR MA780
U 1070, 0810.0038.0180.0968.0000.1341  ;3653 =0000 D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1071, 0810.0038.0180.0968.0000.1341  ;3654  D_RC[0D],J/DONE  ; MS780-A/C
U 1072, 0810.0038.0180.0968.0000.1341  ;3655  D_RC[0D],J/DONE  ; MS780-A/C
U 1073, 0810.0038.0180.0968.0000.1341  ;3656  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1074, 0810.0038.0180.0968.0000.1341  ;3657  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1075, 0810.0038.0180.0968.0000.1341  ;3658  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1076, 0810.0038.0180.0968.0000.1341  ;3659  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1077, 0810.0038.0180.0968.0000.1341  ;3660  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 1078, 0018.0038.0580.0AF0.0000.1079  ;3661  R[0E]_K[.1]  ; SET MA780 FLAG
U 1079, 0810.0038.0180.0968.0000.1341  ;3662  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 107A, 0810.0038.0180.0968.0000.1341  ;3663  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 107B, 0810.0038.0180.0968.0000.1341  ;3664  D_RC[0D],J/DONE  ; NOT A MEMORY, RESTORE TR
U 107C, 0000.003C.0180.0800.0000.11FB  ;3665  J/ENDANY  ; FOUND MS780-E/H
```

```
U 107D. 0000.003C.0180.0800.0000.11FB  ;3666  J/ENDANY  ; FOUND MS780-E/H
U 107E. 0000.003C.0180.0800.0000.11FB  ;3667  J/ENDANY  ; FOUND MS780-E/H
U 107F. 0000.003C.0180.0800.0000.11FB  ;3668  J/ENDANY  ; FOUND MS780-E/H
        ;3669
U 1341. 0019.0020.6180.0800.0010.1342  ;3670 DONE: ALU_D[XOR]K[.F],CLK.UBCC
U 1342. 0000.013C.0180.0800.0000.11EC  ;3671  Z?   ; DONE YET?
U 11EC. 081F.2010.0180.0800.0000.133A  ;3672 =0 J/START,ALU_0+D+1,D_ALU ; NO, INCR TR AND TRY AGAIN
U 11ED. 0000.003C.0180.0A70.0010.1343  ;3673  ALU_R[0E],CLK.UBCC ; WAS THERE AN MA780?
U 1343. 0000.013C.0180.0800.0000.11EE  ;3674  Z?   ; BRANCH IF NO
U 11EE. 0000.003D.0180.0800.0000.1229  ;3675 =0 ERROR1   ; FOUND MA BUT NO MS. WHY ARE YOU
U 11EF. 0000.003C.0180.0800.0000.1344  ;3676  NOP   ; ...RUNNING THIS FLOPPY?
U 1344. 0F00.003C.0180.0800.0000.1345  ;3677  D_0
U 1345. 0000.003C.C180.3C00.0000.11F0  ;3678  ID[T0]_D  ; Set up type out flag
U 11F0. 0000.003D.0180.0800.0000.12BD  ;3679 =0 CALL[DECFLG]
U 11F1. 0000.003C.0180.0800.0000.11F2  ;3680  NOP
U 11F2. 0818.0039.6580.0800.0000.12B9  ;3681 =0 TYP_MSG_10_CRLF  ; " NO MEM CTRLRS"
U 11F3. 0000.003C.0180.0800.0000.1346  ;3682  NOP
U 1346. 0F00.003C.0180.0800.0000.1347  ;3683  D_0
U 1347. 0000.003C.C180.3C00.0000.11F4  ;3684  ID[T0]_D  ; SET UP TYPE OUT FLAG
U 11F4. 0000.003D.0180.0800.0000.12BD  ;3685 =0 CALL[DECFLG]
U 11F5. 0000.003C.0180.0800.0000.11F6  ;3686  NOP
U 11F6. 081B.0011.6580.0800.0000.12B9  ;3687 =0 TYP_MSG_11_CRLF  ; "DEPOSIT CTRLR ADRESS..."
U 11F7. 0000.003C.0180.0800.0000.11F8  ;3688  NOP
U 11F8. 0000.003D.0180.0800.0000.1228  ;3689 =0 ERLOOP
U 11F9. 0010.0038.0180.0900.0200.1348  ;3690  VA_RC[0]
U 1348. 0000.003C.8980.0800.0084.7349  ;3691  SC_K[.D]
U 1349. 0003.001C.0180.0AF8.0000.134A  ;3692  R[0F]_NOT.MASK
U 134A. 081B.001C.C180.0800.0000.134B  ;3693  D_NOT.K[.FFFF]
U 134B. 0F00.003C.6D80.3C00.0000.134C  ;3694  ID[FAULT]_D,D_0
U 134C. 0000.003C.6D80.3C00.0000.134D  ;3695  ID[FAULT]_D
U 134D. 0000.003C.0180.C800.0000.11FA  ;3696  MCT/READ.P
U 11FA. 0000.003D.0180.0800.0000.1229  ;3697 =0 ERROR1
U 11FB. 0000.003C.0180.0800.0000.11FC  ;3698 ENDANY: NOP
            ;3699
            ;3700
```

```
                ;3701  .PAGE "TEST 181 TIMEOUT COUNT TEST"
                ;3702 ;*****************************************************************
                ;3703 ;**
                ;3704 ;
                ;3705 ;TEST 181 TIMEOUT COUNT TEST
                ;3706 ;
                ;3707 ;TEST DESCRIPTION
                ;3708 ; THIS TEST CHECKS THAT THE TIMEOUT COUNTER LOADS AND INCREMENTS
                ;3709 ; CORRECTLY. A NONEXISTANT MEMORY LOCATION IS REFERENCED THUS
                ;3710 ; CAUSING A TIMEOUT TO OCCUR. THE INTERVAL COUNTER IS USED TO
                ;3711 ; COUNT THE CYCLES UNTIL THE TIMEOUT OCCURS. THE INTERVAL COUNTER
                ;3712 ; IS THEN CHEKED TO SEE THAT THE TIMEOUT OCCURRED APPROXIMATELY
                ;3713 ; 512 CYLCES AFTER THE REFERENCE. THE SBI ERROR REGISTER (ID 19)
                ;3714 ; IS THEN CHECKED TO SEE THAT THE TIMEOUT WAS LATCHED (BIT 12).
                ;3715 ; IF THE TIMEOUT WAS LATCHED CORRECTLY, THE REGISTER IS THEN WRITTEN
                ;3716 ; WITH THE VALUE ZERO AND CHECKED TO SEE THAT THE TIMEOUT DOES NOT
                ;3717 ; CLEAR. THE REGISTER IS THEN WRITTEN WITH BIT 12 SET TO SEE THAT
                ;3718 ; THE TIMEOUT CLEARS PROPERLY.
                ;3719 ;
                ;3720 ;LOGIC DESCRIPTION
                ;3721 ; THE TIMEOUT COUNTER ON SBLK AND ID 19 ON SBLM ARE TESTED.
                ;3722 ;
                ;3723 ;ERROR DESCRIPTION
                ;3724 ; EXPECTED INTERVAL COUNTER IN HEX
                ;3725 ; RECEIVED INTERVAL COUNTER IN HEX
                ;3726 ; EXPECTED SBI ERROR REGISTER
                ;3727 ; RECEIVED SBI ERROR REGISTER
                ;3728 ;
                ;3729 ;--
                ;3730 ;*****************************************************************
                ;3731 =0
U 11FC. 0000.003D.0180.0800.0000.1216  ;3732 TMOUT: NEWTST
U 11FD. 0018.0038.1180.09FB.0000.11FE  ;3733  RC[0F]_K[.4]
U 11FE. 0000.003D.0180.0800.0000.1228  ;3734 =0 ERLOOP
U 11FF. 0000.003C.0180.0800.0000.134E  ;3735  NOP
U 134E. 0000.003C.8980.0800.0084.734F  ;3736  SC_K[.D]
U 134F. 0003.001C.0180.0AF8.0000.1350  ;3737  R[0F]_ALU,ALU_NOT.MASK  ;SET TIMEOUT TRAP CATCH BIT
U 1350. 0818.0038.5D80.0800.0000.1351  ;3738  D_K[.7]
U 1351. 0000.003C.85F8.0800.0084.7352  ;3739  SC_K[.C],Q_0
U 1352. 0D00.003C.0180.0800.0000.1353  ;3740  D_DAL.SC
U 1353. 0018.0038.D1C0.0800.0000.1354  ;3741  Q_K[.C0]
U 1354. 081D.0014.0180.0800.0000.1355  ;3742  D_D+Q
U 1355. 0F00.003C.6580.3C0D.0000.1356  ;3743  ID[SBI.ERR]_D,D_0  ;CLR SBI ERROR REGISTER
U 1356. 0000.003C.7580.3C0D.0000.1357  ;3744  ID[MAINT]_D   ;CLR FORCE TIMEOUT IN MAINT REG
U 1357. 0018.0038.F5C0.0800.0000.1358  ;3745  Q_K[.A]
U 1358. 0019.2014.05C0.0800.0000.1359  ;3746  Q_Q+K[.1]
U 1359. 0818.0038.A580.0800.0000.135A  ;3747  D_K[.60]
U 135A. 0019.0014.0180.09FD.0000.135B  ;3748  RC[0E]_D+K[.8]   ;SET MINIMUM EXPECTED INT COUNT
U 135B. 001D.0014.0180.0AD0.0000.135C  ;3749  R[0A]_D+Q   ;SET MAX EXPECTED INTERVAL COUNT
U 135C. 0000.003C.8580.0800.0084.735D  ;3750  SC_K[.C]
U 135D. 0818.0038.0980.0800.0000.135E  ;3751  D_K[.2]
U 135E. 0801.001C.0180.0800.0000.135F  ;3752  D_D.ORNOT.MASK
U 135F. 0001.003C.0180.09E0.0000.1200  ;3753  RC[0C]_D   ;SET EXPECTED SBI ERROR REG
U 1200. 0000.003D.0180.0800.0000.1010  ;3754 =0 CALL,J/TRO   ;GET ADDRESS FOR TIMEOUT
U 1201. 0F01.003C.0180.0800.0200.1360  ;3755  VA_D,D_0   ;VA=20000000
```

```
U 1360. 0000.003C.2580.3C00.0000.1361  ;3756  ID[NXT.PER]_D   ;SET NEXT INTERVAL COUNTER
U 1361. 0818.0038.6580.0800.0000.1362  ;3757  D_K[.10]   ;SET TRANSFER BIT IN
U 1362. 0000.003C.2980.3C00.0000.1363  ;3758  ID[CLK.CS]_D   ;INTERVAL CLK CONTROL REG
U 1363. 0818.0038.0580.0800.0000.1364  ;3759  D_K[.1]   ;SET THE RUN BIT IN
U 1364. 0000.003C.2980.3C00.0000.1365  ;3760  ID[CLK.CS]_D   ;INTERVAL CLK CONTROL REG
U 1365. 0000.003C.0180.C800.0000.1366  ;3761  MCT/READ.P   ;DO THE REFERENCE TO CAUSE TIMEOUT
U 1366. 0F00.003C.0180.0800.0000.1367  ;3762  D_0
U 1367. 0000.003C.2980.3C00.0000.1368  ;3763  ID[CLK.CS]_D   ;CLEAR THE RUN BIT
U 1368. 0000.003C.65F0.2C00.0000.1369  ;3764  Q_ID[SBI.ERR]   ;READ THE ERROR REGISTER
U 1369. 0001.203C.0180.09D8.0000.136A  ;3765  RC[0B]_Q   ;SAVE
U 136A. 0000.003C.2DF0.2C00.0000.136B  ;3766  Q_ID[INTERVAL]   ;READ INTERVAL COUNTER
U 136B. 0001.203C.0180.09E8.0000.136C  ;3767  RC[0D]_Q   ;SAVE
U 136C. 0810.0038.0180.0970.0000.136D  ;3768  D_RC[0E]   ;GET MIN INT COUNTER
U 136D. 001D.0000.0180.0800.0010.136E  ;3769 NXTCNT: ALU_D-Q,CLK.UBCC
U 136E. 0001.013C.0180.09F0.0000.1202  ;3770  Z?,RC[0E]_D
U 1202. 0000.003C.0180.0A50.0000.136F  ;3771 =0 LAB_R[0A],J/CHKMAX
U 1203. 0000.003C.0180.0800.0000.1371  ;3772  J/CHKREG
U 136F. 001C.2000.0180.0800.0010.1370  ;3773 CHKMAX: ALU_LA-D,CLK.UBCC  ;MAX COUNT?
U 1370. 0819.0114.0580.0800.0000.1204  ;3774  Z?,D_D+K[.1]
U 1204. 0000.003C.0180.0800.0000.136D  ;3775 =0 J/NXTCNT
U 1205. 0000.003D.0180.0800.0000.1229  ;3776  ERROR1   ;TIMEOUT COUNT INCORRECT
U 1371. 0810.0038.0180.0958.0000.1372  ;3777 CHKREG: D_RC[0B]   ;GET RECEIVED SBI ERROR REG
U 1372. 0010.0038.01C0.0960.0000.1373  ;3778  Q_RC[0C]   ;GET EXPECTED SBI ERROR REG
U 1373. 001D.0000.0180.0800.0010.1374  ;3779  ALU_D-Q,CLK.UBCC
U 1374. 0000.013C.0180.0800.0000.1206  ;3780  Z?
U 1206. 0000.003D.0180.0800.0000.1229  ;3781 =0 ERROR1   ;ERROR REGISTER DATA INCORRECT
U 1207. 0818.0038.1980.0800.0000.1375  ;3782  D_K[ZERO]
U 1375. 0000.003C.6580.3C00.0000.1376  ;3783  ID[SBI.ERR]_D   ;WRITE ZERO TO SBI ERROR REG
U 1376. 0810.0038.0180.0960.0000.1377  ;3784  D_RC[0C]   ;GET EXPECTED SBI ERROR REG
U 1377. 0000.003C.65F0.2C00.0000.1378  ;3785  Q_ID[SBI.ERR]   ;READ SBI ERROR REG
U 1378. 001D.0000.0180.0800.0010.1379  ;3786  ALU_D-Q,CLK.UBCC
U 1379. 0001.213C.0180.09D8.0000.1208  ;3787  Z?,RC[0B]_Q
U 1208. 0000.003D.0180.0800.0000.1229  ;3788 =0 ERROR1   ;TIMEOUT SHOULD NOT CLEAR
U 1209. 0000.003C.8580.0800.0084.737A  ;3789  SC_K[.C]
U 137A. 0803.001C.0180.0800.0000.137B  ;3790  D_NOT.MASK
U 137B. 0000.003C.6580.3C00.0000.137C  ;3791  ID[SBI.ERR]_D   ;WRITE 1 TO BIT 12 TO CLR T.O.
U 137C. 0818.0038.0980.0800.0000.137D  ;3792  D_K[.2]   ;SET EXPECTED SBI ERROR REG
U 137D. 0000.003C.65F0.2C00.0000.137E  ;3793  Q_ID[SBI.ERR]   ;READ THE SBI ERROR REG
U 137E. 0001.203C.0180.09D8.0000.137F  ;3794  RC[0B]_Q   ;SAVE
U 137F. 001D.0000.0180.0800.0010.1380  ;3795  ALU_D-Q,CLK.UBCC
U 1380. 0001.013C.0180.09E0.0000.120A  ;3796  Z?,RC[0C]_D
U 120A. 0000.003D.0180.0800.0000.1229  ;3797 =0 ERROR1   ;TIMEOUT DID NOT CLEAR
U 120B. 0000.003C.0180.0800.0000.120C  ;3798  NOP
          ;3799
```

```
            ;3800  .PAGE "TEST 182 RESERVED"
            ;3801 =0
U 120C, 0000,003D,0180,0800,0000,1216  ;3802 T182: NEWTST
U 120D, 0000,003C,0180,0800,0000,120E  ;3803  NOP
            ;3804
```

```
            ;3805  .PAGE "TEST 183 RESERVED"
            ;3806 =0
U 120E, 0000,003D,0180,0800,0000,1216  ;3807 T183: NEWTST
U 120F, 0000,003C,0180,0800,0000,1210  ;3808  NOP
            ;3809
```

```
            ;3810  .PAGE "TEST 184 RESERVED"
            ;3811 =0
U 1210. 0000.003D.0180.0800.0000.1216  ;3812 T184: NEWTST
U 1211. 0000.003C.0180.0800.0000.1381  ;3813  NOP
            ;3814
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 0130
ESKAR3E.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:28          Page  11

            ;3815  .PAGE "DUMMY NEWTST"
U 1381,'0000,003D,0180,0800,0000,1216  ;3816 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0131
ESKAR3E.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:28     Page  11
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1920: 1887 1894= 1910= 2238= 2239= 2246= 2248=
U 1008  2337= 2338= 2340= 2342= 2190= 2191= 2192= 2193=
U 1010  2390= 2391= 2392= 2393= 2394= 2395= 2396= 2397=
U 1018  2398= 2399= 2400= 2401= 2402= 2403= 2404= 2405=
U 1020  2735= 2736= 2737= 2738= 2754= 2755= 2756= 2757=
U 1028  2995= 2996= 2997= 1888 1911= 1912= 2578= 2579=
U 1030  3015= 3016= 3017= 1889 1970= 1971= 1890 2639=
U 1038  3088= 3089= 3090= 3091= 1979= 1980= 1891 2640=
U 1040  3299= 3300= 3301= 3302= 2003= 2004= 1892 2856=
U 1048  3329= 3330= 3331= 3332= 2029= 2030= 2011: 2857=
U 1050  3362= 3363= 3364= 3365= 3397= 3398= 3399= 3400=
U 1058  2138= 2139= 2037: 1893 2317= 2318= 2043: 1913
U 1060  2647= 2648= 2649= 2650= 2651= 2653= 2654= 2655=
U 1068  2656= 2657= 2658= 2659= 2660= 2661= 2664= 2665=
U 1070  3653= 3654= 3655= 3656= 3657= 3658= 3659= 3660=
U 1078  3661= 3662= 3663= 3664= 3665= 3666= 3667= 3668=
U 1080  3431= 3432= 3433= 3434= 2335= 2336= 2344= 2345=
U 1088  2358= 2359= 2363= 2364= 2368= 2369= 2425= 2426=
U 1090  2430= 2431= 2451= 2452= 2453= 2454= 2455= 2456=
U 1098  2457= 2458= 2459= 2460= 2478= 2479= 2497= 2498=
U 10A0  2518= 2519= 2521= 2522= 2528= 2529= 2568= 2569=
U 10A8  2572= 2573= 2575= 2576= 2582= 2583= 2586= 2587=
U 10B0  2589= 2590= 2641= 2642= 2674= 2675= 2683= 2684=
U 10B8  2685= 2686= 2697= 2698= 2699= 2700= 2701= 2702=
U 10C0  2703= 2704= 2707= 2708= 2714= 2715= 2723= 2724=
U 10C8  2728= 2729= 2730= 2731= 2732= 2733= 2739= 2740=
U 10D0  2743= 2744= 2745= 2746= 2749= 2750= 2751= 2752=
U 10D8  2758= 2759= 2762= 2763= 2764= 2765= 2776= 2777=
U 10E0  2778= 2779= 2780= 2781= 2782= 2783= 2794= 2795=
U 10E8  2796= 2797= 2798= 2799= 2819= 2820= 2821= 2822=
U 10F0  2823= 2824= 2825= 2826= 2827= 2828= 2887= 2888=
U 10F8  2915= 2916= 2974= 2975= 2982= 2983= 2984= 2985=
U 1100  1874: 1875: 1876: 1877: 1878: 1879: 1880: 1881:
U 1108  1882: 1914 2986= 2987= 1883: 1884: 1885: 1886:
U 1110  2988= 2989= 2990= 2991= 2992= 2993= 3003= 3004=
U 1118  3005= 3006= 3007= 3008= 3009= 3010= 3011= 3012=
U 1120  3013= 3014= 3070= 3071= 3072= 3073= 3074= 3075=
U 1128  3077= 3078= 3079= 3080= 3081= 3082= 3086= 3087=
U 1130  3097= 3098= 3099= 3100= 3101= 3102= 3104= 3105=
U 1138  3106= 3107= 3114= 3115= 3156= 3157= 3158= 3159=
U 1140  3160= 3161= 3162= 3163= 3172= 3173= 3183= 3184=
U 1148  3185= 3186= 3194= 3195= 3200= 3201= 3202= 3203=
U 1150  3204= 3205= 3272= 3273= 1915 2066: 3278= 3279=
U 1158  3280= 3281= 3282= 3283= 3284= 3285= 3286= 3287=
U 1160  3292= 3293= 3295= 3296= 3297= 3298= 3304= 3305=
U 1168  3310= 3311= 3312= 3313= 3314= 3315= 3316= 3317=
U 1170  3318= 3319= 3320= 3321= 3322= 3323= 3325= 3326=
U 1178  3327= 3328= 3334= 3335= 3340= 3341= 3342= 3343=
U 1180  3344= 3345= 3346= 3347= 3348= 3349= 3352= 3353=
U 1188  3355= 3356= 3358= 3359= 3360= 3361= 3367= 3368=
U 1190  3373= 3374= 3375= 3376= 3377= 3378= 3379= 3380=
U 1198  3381= 3382= 3390= 3391= 3393= 3394= 3395= 3396=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  3402= 3403= 3408= 3409= 3410= 3411= 3412= 3413=
U 11A8  3414= 3415= 3416= 3417= 3424= 3425= 3427= 3428=
U 11B0  3429= 3430= 3436= 3437= 3513= 3514= 3516= 3517=
U 11B8  3518= 3519= 3521= 3522= 3523= 3524= 3525= 3526=
U 11C0  3527= 3528= 3530= 3531= 3532= 3533= 3540= 3541=
U 11C8  3544= 3545= 3547= 3548= 3550= 3552= 3559= 3560=
U 11D0  3563= 3564= 3565= 3566= 3573= 3574= 3581= 3582=
U 11D8  3585= 3586= 3587= 3588= 3596= 3597= 3600= 3601=
U 11E0  3602= 3603= 3608= 3609= 3636= 3637= 3640= 3641=
U 11E8  3646= 3647= 3649= 3650= 3672= 3673= 3675= 3676=
U 11F0  3679= 3680= 3681= 3682= 3685= 3686= 3687= 3688=
U 11F8  3689= 3690= 3697= 3698= 3732= 3733= 3734= 3735=
U 1200  3754= 3755= 3771= 3772= 3775= 3776= 3781= 3782=
U 1208  3788= 3789= 3797= 3798= 3802= 3803= 3807= 3808=
U 1210  3812= 3813= 1916 1917 1918 1919 1959 1960
U 1218  1961 1962 1963 1964 1965 1966 1967 1968
U 1220  1969 1972 1973 1974 1975 1976 1977 1978
U 1228  1984 2009 2010 2035 2036 2077 2078 2087
U 1230  2088 2089 2098 2099 2100 2101 2126 2127
U 1238  2128 2129 2130 2131 2132 2133 2134 2136
U 1240  2137 2184 2185 2186 2188 2189 2232 2233
U 1248  2234 2235 2236 2250 2252 2260 2261 2262
U 1250  2269 2270 2271 2284 2285 2286 2287 2288
U 1258  2310 2311 2312 2313 2314 2315 2316 2319
U 1260  2320 2321 2332 2333 2334 2343 2355 2356
U 1268  2357 2360 2361 2362 2365 2366 2367 2406
U 1270  2407 2408 2409 2410 2427 2428 2429 2432
U 1278  2433 2434 2461 2462 2463 2480 2481 2482
U 1280  2492 2493 2494 2495 2496 2520 2523 2524
U 1288  2525 2526 2527 2530 2564 2565 2566 2567
U 1290  2570 2571 2574 2577 2580 2581 2584 2585
U 1298  2588 2591 2592 2593 2635 2636 2637 2638
U 12A0  2645 2646 2687 2688 2689 2690 2706 2726
U 12A8  2727 2773 2067: 2774 2775 2785 2802 2803
U 12B0  2805 2848 2850 2851 2852 2853 2854 2855
U 12B8  2858 2861 2862 2863 2865 2874 2875 2876
U 12C0  2885 2886 2889 2890 2891 2892 2917 2918
U 12C8  2919 2933 2934 2935 2936 2937 2976 2977
U 12D0  2994 3076 3083 3084 3085 3103 3108 3109
U 12D8  3110 3111 3112 3113 3116 3117 3118 3119
U 12E0  3164 3165 3166 3167 3168 3169 3170 3171
U 12E8  3174 3175 3176 3177 3178 3179 3180 3181
U 12F0  3182 3187 3188 3189 3190 3191 3192 3193
U 12F8  3196 3197 3198 3199 3206 3207 3288 3289
U 1300  3290 3291 3294 3303 3324 3333 3350 3351
U 1308  3354 3357 3366 3383 3384 3385 3386 3387
U 1310  3388 3389 3392 3401 3418 3419 3420 3421
U 1318  3422 3423 3426 3435 3515 3529 3534 3536
U 1320  3538 3539 3546 3553 3554 3555 3556 3557
U 1328  3558 3570 3571 3572 3575 3576 3577 3578
U 1330  3579 3580 3591 3592 3593 3594 3595 3605
U 1338  3606 3607 3638 3639 3642 3643 3644 3645
U 1340  3651 3670 3671 3674 3677 3678 3683 3684
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3691 3692 3693 3694 3695 3696 3736 3737
U 1350  3738 3739 3740 3741 3742 3743 3744 3745
U 1358  3746 3747 3748 3749 3750 3751 3752 3753
U 1360  3756 3757 3758 3759 3760 3761 3762 3763
U 1368  3764 3765 3766 3767 3768 3769 3770 3773
U 1370  3774 3777 3778 3779 3780 3783 3784 3785
U 1378  3786 3787 3790 3791 3792 3793 3794 3795
U 1380  3796 3816
U 1388 - 13EF Unused
U 13F0  2050: 2051: 2052: 2053: 2054: 2055: 2056: 2057:
U 13F8  2058: 2059: 2060: 2061: 2062: 2063: 2064: 2065:
```

F 11

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0134
ESKAR3E.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:28     Page  12

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR3E        914

Used              914
Remaining

Total microwords used in memory U: 914
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0135
ESKAR3E.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:28     Page  12


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR3E.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR3F.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 0136

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2173 DIAGNOSTIC SPECIFIC SUBROUTINES
; 2174   SELECT CONTROLLER
; 2212   64K OR 256K CHIPS
; 2266   START TEST
; 2475   SELECT LOWER CONTROLLER
; 2523   SELECT UPPER CONTROLLER
; 2571   SET DIAGNOSTIC MODE ROUTINE
; 2600   SBI UNJAM ROUTINE
; 2649   RESET SUBTEST NUMBER
; 2671   INITIALIZE THE SBI
; 2698   DISABLE CACHE ROUTINE
; 2719   ENABLE CACHE
; 2740   WAIT LOOP ROUTINE
; 2773   CHECK FOR INTERRUPT ROUTINE
; 2820   CHECK UTRAP
; 2852   CHECK SBI.ERR
; 2882   SET TRAP MASK
; 2910   FIND MS780-E MEMORY
; 3052   GENERATE IO ADDRESS
; 3077   SET STARTING ADDRESS
; 3110   ENABLE NEXT MS780-E
; 3162   LOAD TEMPORARY REGISTERS
; 3204   REGISTER DATA INTEGRITY CHECK
; 3383   GET MEMORY SIZE
; 3442 TEST 185 FORCE TIME-OUT TEST
; 3534 TEST 186 CNFG REGISTER ACCESS
; 3775 TEST 187 DATA LATCHES 1 AND 2 TEST
; 3885 TEST 188 STARTING ADDRESS TEST
; 4218 TEST 189 CONTROLLER LATCHES
; 4570 TEST 18A BASIC MEMORY WRITE READ
; 5016 TEST 18B RESERVED
; 5021 TEST 18C RESERVED
; 5027 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        :1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR3F
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
;1841  .PAGE
;1842
;1843  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844  .TOC "   Micro Trap Handler"
;1845 ;++
;1846 ; FUNCTIONAL DESCRIPTION:
;1847 ;
;1848 ; This routine is responsible for 'catching' micro-traps
;1849 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850 ; contains the Trap-Expected Mask. If the specified bit is set in
;1851 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852 ; an error call is made.
;1853 ;
;1854 ; INPUT PARAMETERS:
;1855 ;
;1856 ; R[0F] - Expected Trap Mask
;1857 ;  Trap Code Function
;1858 ;  --------      --------
;1859 ;       0  System Init
;1860 ;       1  Data Unaligned
;1861 ;       2  Cross Page
;1862 ;       3  TB Modify
;1863 ;       4  TB Protection
;1864 ;       6  TB Miss
;1865 ;       7  TB Parity
;1866 ;       8  Cache Parity
;1867 ;       9  Reserved Floating Operand
;1868 ;       C  Read Data Substitute
;1869 ;       D  Time out
;1870 ;       F  Control Store Parity Error
;1871 ;      10  Odd Address
;1872 ;
;1873 ; OUTPUT PARAMETERS:
;1874 ;
;1875 ; R[0F] - Mask bit is cleared.
;1876 ;
;1877 ; DATA: Micro PC of Micro Trap
;1878 ;  Trap Code (See Above)
;1879 ;  Fault Status Register
;1880 ;  SBI Error Register
;1881 ;  Timeout Address Register
;1882 ;  Maintenance Register
;1883 ;  Cache Parity Register
;1884 ;  TB Error Register 1
;1885 ;  TB Error Register 0
;1886 ;--
U 1100,  0000.003C.0180.0800.0084.7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101,  0000.003C.0580.0800.0084.7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102,  0000.003C.0980.0800.0084.7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103,  0000.003C.0D80.0800.0084.7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104,  0000.003C.1180.0800.0084.7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105,  0000.003C.D580.0800.0084.7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106,  0000.003C.D980.0800.0084.7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107,  0000.003C.5D80.0800.0084.7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108,  0000.003C.0180.0800.0084.7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000.003C.8580.0800.0084.7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000.003C.8980.0800.0084.7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000.003C.6580.0800.0084.7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000.003C.6180.0800.0084.7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1900
U 1001, 0000.003C.81F0.2C00.0000.1047  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 1047, 0C00.003C.01E0.0800.0000.104F  ;1902  q_d,d_q   ; Setup to put it back on stack
U 104F, 0C00.003C.81E0.3C00.0000.1053  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1053, 0000.003C.01C0.0A78.0000.105B  ;1904  q_r[0f]   ; Get the Expected Trap Mask
        ;1905  alu_q.andnot.mask,
U 105B, 0001.2024.0180.0800.0010.105F  ;1906   clk.ubcc ; Was trap expected?
U 105F, 0000.013C.0180.0800.0000.1002  ;1907  z?
        ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
        ;1909       alu_q.and.mask, ; ...
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1910       return0  ; ...
U 1003, 0018.0038.1D80.09E8.0000.104C  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 104C, 0000.003D.D980.0800.0084.71D5  ;1912 =0 setrcf[.9]  ; Set output count
U 104D, 0000.003C.49F0.2C00.0000.106B  ;1913  q_id[tber0]  ; Output all the error registers
U 106B, 0001.203C.4DF0.2DB0.0000.106F  ;1914  rc[6]_q,q_id[tber1] ; ...
U 106F, 0001.203C.65F0.2DB8.0000.1073  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 1073, 0001.203C.6DF0.2DD8.0000.1077  ;1916  rc[0b]_q,q_id[fault] ; ...
U 1077, 0001.203C.75F0.2DE0.0000.107B  ;1917  rc[0c]_q,q_id[maint] ; ...
U 107B, 0001.203C.79F0.2DC8.0000.1087  ;1918  rc[9]_q,q_id[parity] ; ...
U 1087, 0001.203C.69F0.2DC0.0000.108B  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 108B, 0001.203C.81F0.2DD0.0000.1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
        ;1921 1000: ERROR1[.C],  ;
U 1000, 0001.203D.8580.09F0.0084.70C0  ;1922   rc[0e]_q ; Report the error
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "   Newtest Routine"
                ;1926 ;++
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to    0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 10A3. 0000,003C,65F8,0800,0084,70A7   ;1954 SCOPE: sc_k[.10],q_0   ; Set IPL to 1F
U 10A7. 0818,0038,8D80,0800,0000,1109   ;1955  d_k[.1f]   ; D=1F
U 1109. 0D00,003C,0180,0800,0000,11BE   ;1956  d_dal.sc   ; D=001F0000
U 11BE. 0000,003C,3D80,3C00,0000,11BF   ;1957  id[psl]_d   ; psl = 001F0000
U 11BF. 0818,0038,0580,0800,0000,11C0   ;1958  d_k[.1]    ; Clear the CES register
U 11C0. 0D00,003C,0180,0800,0000,11C1   ;1959  d_dal.sc   ; D = 00010000
U 11C1. 0F00,003C,3180,3C00,0000,11C2   ;1960  id[ces]_d,d_0   ; ces = 00010000
U 11C2. 0000,003C,5D80,3C00,0000,11C3   ;1961  id[acc.cs]_d   ; acc.cs = 0
U 11C3. 0000,003C,3980,3C00,0000,11C4   ;1962  id[sir]_d   ; sir = 0
U 11C4. 0000,003C,2980,3C00,0000,11C5   ;1963  id[clk.cs]_d   ; clk.cs = 0
U 11C5. 0000,003C,4980,3C00,0000,1058   ;1964  id[tber0]_d   ; tber0 = 0
U 1058. 0000,003D,0180,0800,0000,11F6   ;1965 =0 call[sbi.unjam]   ; Unjam the SBI
                ;1966  intrpt.strobe,   ; Strobe interrupts
U 1059. 0818,0038,C180,0800,4000,11C6   ;1967       d_k[.ffff]   ; Setup to clear SBI error register and
U 11C6. 0801,0028,6580,3C00,0000,11C7   ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 11C7. 0F00,003C,6D80,3C00,0000,11C8   ;1969  id[fault]_d,d_0   ; ...
U 11C8. 0000,003C,6D80,3C00,0000,11C9   ;1970  id[fault]_d   ; fault = 0
U 11C9. 0000,003C,7580,3C00,0000,11CA   ;1971  id[maint]_d   ; MAINT = 0
U 11CA. 0000,003C,6580,3C00,0000,11CB   ;1972  id[sbi.err]_d   ; sbi error reg = 0
U 11CB. 0000,003C,7180,3C00,0000,11CC   ;1973  id[comp]_d   ; comp reg = 0
U 11CC. 0000,003C,E580,3C00,0000,11CD   ;1974  ID[39]_d    ; Clear code for subroutine START.TEST
U 11CD. 0001,003C,0180,09F0,0000,11CE   ;1975  rc[0e]_d   ;
U 11CE. 0001,003C,0180,0AF8,0000,11CF   ;1976  r[0f]_d    ; Clear expected trap mask
U 11CF. 0001,003C,0180,09F8,0000,105C   ;1977  rc[0f]_d   ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,11FC  ;1978 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1980  .PAGE
            ;1981  .TOC "   Error Loop Routine"
            ;1982 ;++
            ;1983 ; FUNCTIONAL DESCRIPTION:
            ;1984 ;
            ;1985 ; This routine is called with the "ERLOOP" macro. The
            ;1986 ; routine calls the Micro Monitor to setup an error loop.
            ;1987 ;
            ;1988 ; CALLING CONSTRAINT:
            ;1989 ;
            ;1990 ; =0
            ;1991 ;
            ;1992 ; SIDE EFFECTS:
            ;1993 ;
            ;1994 ; D Reg - Destroyed
            ;1995 ;--
U 11D0, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 11D1, 0810,0038,0180,0978,0000,11D2  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 11D2, 0819,0034,4D80,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;++
            ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 10C0, 0000,003D,0180,0800,0000,11D5  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10C1, 0000,003C,0180,0800,0000,11D1  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "   Error 2 Routine"
            ;2045 ;**
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 11D3, 0810,0038,0180,0978,0000,11D4  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 11D4, 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "   ERROR 2 WITH PARAMETER"
            ;2071 ;++
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 10C2, 0000,003D,0180,0800,0000,11D5  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10C3, 0000,003C,0180,0800,0000,11D3  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0149
ESKAR3F.MCR                    MICRO2  1P(00)    26-JUL-85  17:04:54                     Page   6

            ;2088  .PAGE
            ;2089  .TOC "    Micro-Monitor Call"
            ;2090 ;*
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;++
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2111 13F0: nop
U 13F1, 0000.003C.0180.0800.0000.13F2  ;2112 13F1: nop
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2113 13F2: nop
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2114 13F3: nop
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2115 13F4: nop
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2116 13F5: nop
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2117 13F6: nop
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2118 13F7: nop
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2119 13F8: nop
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2120 13F9: nop
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2121 13FA: nop
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2122 13FB: nop
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2123 13FC: nop
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2124 13FD: nop
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2125 13FE: nop
U 13FF, 0000.003C.0180.0800.0000.12AA  ;2126 13FF: nop
U 12AA, 0000.003C.0180.0800.0000.11D5  ;2127 12AA: nop ; This location is permenently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;**
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 11D5, 0010,0038,01C0,0978,0000,11D6  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 11D6, 0019,2034,4DC0,0800,0000,11D7  ;2150  q_q.and.k[.ff00] ; ...
U 11D7, 0019,2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "   Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 11D8, 0810,0038,4D80,0978,0000,11D9  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 11D9, 0019,4016,4D80,09F8,0000,0001  ;2171       word,return1 ; (Subtest number is negative)
```

```
            ;2172  .PAGE
            ;2173  .TOC " DIAGNOSTIC SPECIFIC SUBROUTINES"
            ;2174  .TOC "   SELECT CONTROLLER"
            ;2175 ;**
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine is used to put the memory system into the
            ;2179 ; specified configuration with respect to controller select.
            ;2180 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2181 ; a RETURN2 is made.
            ;2182 ;
            ;2183 ; CALLING CONSTRAINT:
            ;2184 ;
            ;2185 ; =00
            ;2186 ;
            ;2187 ; INPUTS:
            ;2188 ;
            ;2189 ; d - Contains value to write to bits<2:0> of Register A
            ;2190 ; rc[0e] - Address of Register A
            ;2191 ;
            ;2192 ; SIDE EFFECTS:
            ;2193 ;
            ;2194 ; sc,d,va are destroyed
            ;2195 ;--
            ;2196 SELECT.CNTRLR:
U 11DA, 0010.0038.0180.0970.0284.71DB  ;2197  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11DB, 0801.001C.0180.0800.0000.11DC  ;2198  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11DC, 0000.003C.0180.A800.0000.11DD  ;2199  cache.p_d[long]  ; Write the configuration Register
U 11DD, 0818.0038.5D80.0800.0000.11DE  ;2200  d_k[.7]  ; Get Misconfiguration bits
U 11DE, 0000.003C.61F8.0800.0084.71DF  ;2201  sc_k[.F],q_0  ; ...
U 11DF, 0D00.003C.0180.0800.0000.11E0  ;2202  d_dal.sc  ; ... D = x38000
U 11E0, 0000.003C.01E0.0800.0000.11E1  ;2203  q_d  ; Save mask
U 11E1, 0000.003C.0180.C800.0000.11E2  ;2204  d[long]_cache.p  ; Read CNFG A Reg and
            ;2205  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11E2, 001D.2034.0180.0800.0010.11E3  ;2206  clk.ubcc ; ...
U 11E3, 0000.013C.0180.0800.0000.10C4  ;2207  z?  ; Branch if no
U 10C4, 0000.003E.0180.0800.0000.0001  ;2208 =0 RETURN1  ; Bad configuration
U 10C5, 0000.003E.0180.0800.0000.0002  ;2209  RETURN2  ; Configuration ok
            ;2210
```

```
            ;2211  .PAGE
            ;2212  .TOC "    64K OR 256K CHIPS"
            ;2213 ;*****************************************************
            ;2214 ;**
            ;2215 ;
            ;2216 ; Functional Description:
            ;2217 ; ----------------------
            ;2218 ; This subroutine is used to find the type of chips
            ;2219 ; on the MS780-E/H arrays.  The RETURN modes are as
            ;2220 ; follows:
            ;2221 ;
            ;2222 ; RETURN1 = Error. Mixed 64K and 256K arrays
            ;2223 ;
            ;2224 ; RETURN2 = 64K chips on the array
            ;2225 ;
            ;2226 ; RETURN3 = 256K chips on the array
            ;2227 ;
            ;2228 ; Calling Constraint: =00
            ;2229 ; ------------------
            ;2230 ;
            ;2231 ;
            ;2232 ; Inputs:
            ;2233 ; ------
            ;2234 ;
            ;2235 ; RC[0E] must hold the I/O base address of the MS780-E/H
            ;2236 ; currently under test
            ;2237 ;
            ;2238 ; Outputs:
            ;2239 ; -------
            ;2240 ;
            ;2241 ; NO specific outputs. (See RETURN modes above)
            ;2242 ;
            ;2243 ;
            ;2244 ; Side Effects:
            ;2245 ; ------------
            ;2246 ;
            ;2247 ; The contents of the following registers will be lost:
            ;2248 ;
            ;2249 ; D
            ;2250 ;
            ;2251 ;--
            ;2252 ;*****************************************************
            ;2253 64K.OR.256K:
U 11E4, 0010,0038,0180,0970,0200,11E5  ;2254  VA_RC[0E]    ; Point to CNFG Register A
U 11E5, 0000,003C,0180,C800,0000,11E6  ;2255  D[LONG]_CACHE.P    ; Read the register
U 11E6, 0200,003C,0180,0800,0000,11E7  ;2256  D_D.RIGHT2    ; Shift received contents two bits
            ;2257          ; ...to the right
U 11E7, 0600,003C,0180,0800,0000,11E8  ;2258  D_D.RIGHT    ; Shift one more time
U 11E8, 0000,193C,0180,0800,0000,100C  ;2259  D1-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2260 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2261  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2262  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2263  RETURN1     ; Error: missing arrays
            ;2264
```

```
;2265  .PAGE
;2266  .TOC "   START TEST"
;2267 ;*************************************************************
;2268 ;**
;2269 ;
;2270 ; Functional Description:
;2271 ; ---------------------
;2272 ;
;2273 ;        This subroutine will be called by all tests that  check  the
;2274 ;    controllers,  or  those  tests  that  have  to switch between the
;2275 ;    controllers when executing.   Its  main  functions  are:   select
;2276 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2277 ;    test that called it, call out an error  if  neither  one  of  the
;2278 ;    controllers can be configured properly, if all the controllers on
;2279 ;    a MS780-E/H were configured and tested, it should select the next
;2280 ;    MS780-E/H on the SBI and repeat the above sequence.
;2281 ;    A test making use of this  subroutine  should  be  configured  as
;2282 ;    follows:
;2283 ;
;2284 ;
;2285 ;       Begin TEST.XX
;2286 ;        :
;2287 ;        :   Call NEWTST
;2288 ;        :   Call FIND.MS780E
;2289 ;        :   IF No MS780-E's on the SBI
;2290 ;        :     THEN
;2291 ;        :         Skip to End of TEST.XX
;2292 ;        :     ELSE
;2293 ;        :
;2294 ;        : RESTART.TEST.XX:
;2295 ;        :
;2296 ;        :   Call START TEST
;2297 ;        :   IF Return1 from START.TEST
;2298 ;        :     THEN
;2299 ;        :         Skip to End of TEST.XX
;2300 ;        :     ELSE.
;2301 ;        :
;2302 ;        ~
;2303 ;
;2304 ;        :
;2305 ;        :   Body of TEST.XX
;2306 ;        :
;2307 ;        ~
;2308 ;
;2309 ;        :   Jump RESTART.TEST.XX
;2310 ;        :
;2311 ;       End TEST.XX
;2312 ;
;2313 ;
;2314 ;          This subroutine makes use of a pointer  in  ID  Register  39
;2315 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2316 ;    should be passed when the subroutine is called a second, third or
;2317 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2318 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2319 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2320 ;    Register 39 can be interpreted as follows:
;2321 ;
;2322 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2323 ;               called the first time.  When this code is encountered
;2324 ;               this subroutine will try to select the lower controller.
;2325 ;               If the lower controller is misconfigured, the pointer in
;2326 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2327 ;               is re-entered.
;2328 ;               If, however there is no misconfiguration, the value in
;2329 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2330 ;               calling test.
;2331 ;
;2332 ;     b01   -   This value signifies that an attempt at configuring the
;2333 ;               lower controller failed, and the subroutine will attempt
;2334 ;               to select the upper controller.
;2335 ;               If the upper controller is also misconfigured execution
;2336 ;               stops with a call to ERROR1.
;2337 ;               If there is no misconfiguration on this controller, then
;2338 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2339 ;               made to the calling test.
;2340 ;
;2341 ;     b10   -   This value signifies that the lower controller was selected
;2342 ;               previously and the test was executed once.  If the
;2343 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2344 ;               39 is loaded with b11 and an attempt is made to select the
;2345 ;               upper controller.
;2346 ;               If there is a misconfiguration on this controller, this
;2347 ;               subroutine is just re-entered.
;2348 ;               Otherwise, a Return2 will be made to the calling test.
;2349 ;
;2350 ;     b11   -   This code identifies the cases in which the test has
;2351 ;               been executed at least once (i.e., at least one of the
;2352 ;               controllers on the MS780-E unit under test could be
;2353 ;               configured properly).  When this code is detected the
;2354 ;               subroutine clears ID Reg 39 and makes a call to
;2355 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2356 ;               tested then the subroutine executes a Return1.
;2357 ;               Otherwise the subroutine is re-entered.
;2358 ;
;2359 ;
;2360 ; Calling Constraint: =00
;2361 ; ------------------
;2362 ;
;2363 ;
;2364 ; Inputs:
;2365 ; ------
;2366 ;
;2367 ; ID Register 39 must be cleared by NEWTST
;2368 ;
;2369 ;
;2370 ; Outputs:
;2371 ; -------
;2372 ;
;2373 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2374 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2375 ;         to be tested
            ;2376 ;
            ;2377 ; Side Effects:
            ;2378 ; ------------
            ;2379 ;
            ;2380 ; Contents of the following regiters will be destroyed:
            ;2381 ;
            ;2382 ; D, Q
            ;2383 ;
            ;2384 ;--
            ;2385 ;*************************************************************
            ;2386 =0
            ;2387 START.TEST:
U 10C6, 0000,003D,0180,0800,0000,11F7  ;2388  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 10C7, 0000,003C,0180,0800,0000,10C8  ;2389  NOP    ; ...tests that have more than one
            ;2390 =0      ; ...subtest.)
U 10C8, 0000,003D,0180,0800,0000,11D0  ;2391  ERLOOP    ; Set up the error loop for the
            ;2392       ; ...eventuality that the MS780-E/H
            ;2393       ; ...currently under test has no
            ;2394       ; ...Controllers
            ;2395 RE.ENTRY:    ; Re entry point for this routine
U 10C9, 0000,003C,E5F0,2C00,0000,11E9  ;2396  Q_ID[39]    ; Get the code
U 11E9, 0C00,003C,0180,0800,0000,11EA  ;2397  D_Q    ; Put it in the D Register
U 11EA, 0000,193C,0180,0800,0000,101C  ;2398  DI-0?    ; Branch on the code
U 101C, 0000,003C,0180,0800,0000,1008  ;2399 =1100 J/STRT.CASE00    ; Code is 00
U 101D, 0000,003C,0180,0800,0000,1010  ;2400  J/STRT.CASE01    ; Code is 01
U 101E, 0000,003C,0180,0800,0000,11EF  ;2401  J/STRT.CASE10    ; Code is 10
U 101F, 0000,003C,0180,0800,0000,1017  ;2402  J/STRT.CASE11    ; Code is 11
            ;2403 ;
            ;2404 ;
            ;2405 ; At this point the code in ID[39] was 00
            ;2406 ;
            ;2407 ;
            ;2408 =00
            ;2409 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1020  ;2410  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2411  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,11EB  ;2412   D_K[.1]    ; Set up the code for re entry to this
            ;2413       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2414  D_K[.2]    ; Set up the code for a next call to
            ;2415       ; ...START.TEST indicating that the
            ;2416       ; ...Lower Controller was configured
            ;2417       ; ...( 10 )
            ;2418  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2419   RETURN2    ; Let the test know that the Lower
            ;2420       ; ...has been configured
            ;2421 STRT.LOW.MIS:
            ;2422  ID[39]_D,    ; Save code in ID[39] signifing that
U 11EB, 0000,003C,E580,3C00,0000,10C9  ;2423   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2424       ; ...and re enter this subroutine
            ;2425
            ;2426 ;
            ;2427 ;
            ;2428 ; At this point the code is 01
            ;2429 ;
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                                    SEQ 0158
ESKAR3F.MCR                    MICRO2  1P(00)    26-JUL-85  17:04:54                                     Page   6

            ;2430 ;
            ;2431 =00
            ;2432 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1024  ;2433  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2434  J/STRT.UPR.MIS,    ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,11EC  ;2435   D_0    ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2436  D_K[.3]    ; Upper controller was configured
            ;2437       ; ...thus the code must be changed
            ;2438       ; ...accordingly ( 11 )
            ;2439  ID[39]_D,    ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2440   RETURN2    ; Let the test know that the Upper
            ;2441       ; ...Controller has been configured
            ;2442 STRT.UPR.MIS:
U 11EC, 0000,003C,E580,3C00,0000,11ED  ;2443  ID[39]_D    ; Save the Code ( 00 )
U 11ED, 0018,0038,0D80,09E8,0000,11EE  ;2444  RC[0D]_K[.3]    ; Set up a code for the Fail Chain
U 11EE, 0000,003D,0980,0800,0084,70C0  ;2445  ERROR1[.2]    ; Flag the error.
            ;2446 ;
            ;2447 ;
            ;2448 ; At this point the code is 10
            ;2449 ;
            ;2450 ;
            ;2451 STRT.CASE10:
U 11EF, 0818,0038,0D80,0800,0000,1014  ;2452  D_K[.3]    ; Set the code to 11
            ;2453
            ;2454
            ;2455 =00 ID[39]_D,    ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1024  ;2456   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,10C9  ;2457  J/RE.ENTRY    ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2458  RETURN2    ; Upper Controller configured
            ;2459       ; ...let the test know it
            ;2460 ;
            ;2461 ;
            ;2462 ; At this point the code is 11
            ;2463 ;
            ;2464 ;
            ;2465 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1018  ;2466  D_0    ; Clear the code
            ;2467 =00 ID[39]_D,    ; Save the code
U 1018, 0000,003D,E580,3C00,0000,1251  ;2468   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2469       ; ...on the SBI
U 1019, 0000,003C,0180,0800,0000,10C9  ;2470  J/RE.ENTRY    ; Found another MS780-E/H.   Go and
            ;2471       ; ...attempt to configure the cntrllrs
U 101A, 0000,003E,0180,0800,0000,0001  ;2472  RETURN1    ; No more MS780-E/Hs found
U 101B, 0000,003C,0180,0800,0000,1020  ;2473  NOP
```

```
;2474  .PAGE
;2475  .TOC "   SELECT LOWER CONTROLLER"
;2476 ;*************************************************************
;2477 ;++
;2478 ;
;2479 ; Functional Description:
;2480 ; ----------------------
;2481 ;
;2482 ; This subroutine can be called to select the lower
;2483 ; Controller on a MS780-E/H.
;2484 ; If the Lower controller is misconfigured then a
;2485 ; RETURN1 will be made.  Otherwise a RETURN2
;2486 ;
;2487 ; Calling Constraint: =00
;2488 ; -----------------
;2489 ;
;2490 ;
;2491 ; Inputs:
;2492 ; ------
;2493 ;
;2494 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2495 ;
;2496 ; Outputs:
;2497 ; -------
;2498 ;
;2499 ; RC[0D] will have the address of CNFG Register C
;2500 ;        ( 2000x008 )
;2501 ;
;2502 ; Side Effects:
;2503 ; ------------
;2504 ;
;2505 ; Contents of the following registers will lost:
;2506 ;
;2507 ; D
;2508 ;
;2509 ; The Lower controller on the MS780-E/H will be configured
;2510 ; when the subroutine is exitted with a RETURN2
;2511 ;--
;2512 ;*************************************************************
;2513 =00
;2514 SELECT.LOWER:
;2515  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1020, 0818.0039.1980.0800.0000.11DA  ;2516   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000.003E.0180.0800.0000.0001  ;2517  RETURN1     ; Lower Controller Misconfigured
U 1022, 0810.0038.0180.0970.0000.1023  ;2518  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2519  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1023, 0019.0016.0180.09E8.0000.0002  ;2520   RETURN2    ; Let caller know that the controller
;2521        ; ...was configured properly
```

```
;2522  .PAGE
;2523  .TOC "    SELECT UPPER CONTROLLER"
;2524 ;*************************************************************
;2525 ;**
;2526 ;
;2527 ; Functional Description:
;2528 ; ----------------------
;2529 ;
;2530 ; This subroutine can be called to select the upper
;2531 ; Controller on a MS780-E/H.
;2532 ; If the Upper controller is misconfigured then a
;2533 ; RETURN1 will be made.  Otherwise a RETURN2
;2534 ;
;2535 ; Calling Constraint: =00
;2536 ; ------------------
;2537 ;
;2538 ;
;2539 ; Inputs:
;2540 ; ------
;2541 ;
;2542 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2543 ;
;2544 ; Outputs:
;2545 ; -------
;2546 ;
;2547 ; RC[0D] will have the address of CNFG Register D
;2548 ;        ( 2000x010 )
;2549 ;
;2550 ; Side Effects:
;2551 ; ------------
;2552 ;
;2553 ; Contents of the following registers will lost:
;2554 ;
;2555 ; D
;2556 ;
;2557 ; The Upper controller on the MS780-E/H will be configured
;2558 ; when the subroutine is exitted with a RETURN2
;2559 ;--
;2560 ;*************************************************************
;2561 =00
;2562 SELECT.UPPER:
;2563  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,0980,0800,0000,11DA  ;2564   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2565  RETURN1    ; Upper Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2566  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2567  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,8580,09E8,0000,0002  ;2568   RETURN2   ; Let caller know that the controller
;2569       ; ...was configured properly
```

```
            ;2570  .PAGE
            ;2571  .TOC "   SET DIAGNOSTIC MODE ROUTINE"
            ;2572 ;**
            ;2573 ; FUNCTIONAL DESCRIPTION:
            ;2574 ;
            ;2575 ; This routine is used to set the specified diagnostic mode
            ;2576 ; in Register B. Other bits in the register remain unchanged.
            ;2577 ;
            ;2578 ; CALLING CONSTRAINT:
            ;2579 ;
            ;2580 ; =0
            ;2581 ;
            ;2582 ; INPUTS:
            ;2583 ;
            ;2584 ; d - Contains the mode to set in bits<1:0>
            ;2585 ; rc[0e] - Contains base address of controller
            ;2586 ;
            ;2587 ; SIDE EFFECTS:
            ;2588 ;
            ;2589 ; d,va,sc are destroyed
            ;2590 ;--
            ;2591 SET_DIAG_MODE:
U 11F0, 0010,0038,D9F8,0970,0284,71F1  ;2592  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11F1, 0D00,003C,0180,0803,0000,11F2  ;2593  va_va+4,d_dal.sc ; Shift specified mode into position
U 11F2, 0000,003C,01E0,0800,0000,11F3  ;2594  q_d    ; Save the diagnostic mode bits
U 11F3, 0000,003C,0180,C800,0000,11F4  ;2595  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11F4, 081D,0030,0180,0800,0000,11F5  ;2596  d_d.or.q  ; Set the diagnostic mode bits
U 11F5, 0000,003E,0180,A800,0000,0001  ;2597  cache.p_d[long],return1 ; Set the specified mode and return
            ;2598
```

```
            ;2599  .PAGE
            ;2600  .TOC "   SBI UNJAM ROUTINE"
            ;2601 ;++
            ;2602 ; FUNCTIONAL DESCRIPTION:
            ;2603 ;
            ;2604 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2605 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2606 ; and finally HOLD for the last 16 cycles.
            ;2607 ;
            ;2608 ; CALLING CONSTRAINT:
            ;2609 ;
            ;2610 ; =0
            ;2611 ;
            ;2612 ; SIDE EFFECTS:
            ;2613 ;
            ;2614 ; D Reg destroyed
            ;2615 ;--
            ;2616    ;
            ;2617    ; assert hold for 16 cycles
            ;2618    ;
            ;2619 SBI.UNJAM:
            ;2620  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11F6, 0818,0038,6580,8000,0010,10CA  ;2621       clk.ubcc  ; set micro cc's for next cycle
            ;2622 =0
            ;2623 SBI.UNJAM.1:
            ;2624  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2625       clk.ubcc,z?, ; ...
U 10CA, 0819,0100,0580,8000,0010,10CA  ;2626       j/sbi.unjam.1 ; test for completion
            ;2627    ;
            ;2628    ; assert hold and unjam for 16 cycles
            ;2629    ;
            ;2630  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10CB, 0818,0038,6580,8800,0010,10CC  ;2631       d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2632 =0
            ;2633 SBI.UNJAM.2:
            ;2634  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2635       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10CC, 0819,0100,0580,8800,0010,10CC  ;2636       z?,j/sbi.unjam.2 ; ...
            ;2637    ;
            ;2638    ; assert hold for 16 cycles
            ;2639    ;
            ;2640  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10CD, 0818,0038,6580,8000,0010,10CE  ;2641       clk.ubcc  ; and set cc's for next cycle
            ;2642 =0
            ;2643 SBI.UNJAM.3:
            ;2644  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2645       clk.ubcc,z?, ; ...
U 10CE, 0819,0100,0580,8000,0010,10CE  ;2646       j/sbi.unjam.3 ; ...
U 10CF, 0000,003E,0180,0800,00D0,0001  ;2647  return1   ; exit
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0163
ESKAR3F.MCR                    MICRO2  1P(00)    26-JUL-85  17:04:54                   Page   7

            ;2648  .PAGE
            ;2649  .TOC "   RESET SUBTEST NUMBER"
            ;2650 ;**
            ;2651 ; FUNCTIONAL DESCRIPTION:
            ;2652 ;
            ;2653 ; Since some  of the tests will have to be restarted when two
            ;2654 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2655 ; the subtest counter to zero ( only for thoes tests that have
            ;2656 ; more than one subtest). This subroutine may also be necessary
            ;2657 ; when a test is being loop on.
            ;2658 ;
            ;2659 ; CALLING CONSTRAINT:
            ;2660 ;
            ;2661 ; =0
            ;2662 ; SIDE EFFECTS:
            ;2663 ;
            ;2664 ; D is destroyed
            ;2665 ;
            ;2666 ;--
            ;2667 RESET.SUBTST:
            ;2668  RC[0F]_0,    ; Reset subtest number
U 11F7, 0003,003E,0180,09F8,0000,0001  ;2669   RETURN1    ; ...and return
```

```
            ;2670  .PAGE
            ;2671  .TOC "   INITIALIZE THE SBI"
            ;2672 ;++
            ;2673 ; FUNCTIONAL DESCRIPTION:
            ;2674 ;
            ;2675 ; This routine performs the following functions:
            ;2676 ;
            ;2677 ;  Executes an SBI Unjam
            ;2678 ;  Clears the SBI Fault Latch
            ;2679 ;  Clears the SBI Error Register
            ;2680 ;
            ;2681 ; CALLING CONSTRAINT:
            ;2682 ;
            ;2683 ; =0
            ;2684 ;
            ;2685 ; SIDE EFFECTS:
            ;2686 ;
            ;2687 ; D & Q Register destroyed
            ;2688 ;--
            ;2689 =0
            ;2690 SBI.INIT:
U 10D0, 0000,003D,0180,0800,0000,11F6  ;2691  call[sbi.unjam]  ; Unjam the SBI
U 10D1, 0818,0038,C180,0800,0000,11F8  ;2692  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11F8, 0801,0028,6580,3C00,0000,11F9  ;2693  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11F9, 0F00,003C,6D80,3C00,0000,11FA  ;2694  id[fault]_d,d_0
U 11FA, 0000,003C,6D80,3C00,0000,11FB  ;2695  id[fault]_d
U 11FB, 0000,003E,6580,3C00,0000,0001  ;2696  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2697  .PAGE
            ;2698  .TOC "   DISABLE CACHE ROUTINE"
            ;2699 ;**
            ;2700 ; FUNCTIONAL DESCRIPTION:
            ;2701 ;
            ;2702 ; This routine disables cache hits by setting bits <16:15>
            ;2703 ; in the maintenance register.
            ;2704 ;
            ;2705 ; CALLING SEQUENCE:
            ;2706 ;
            ;2707 ; =0
            ;2708 ;
            ;2709 ; SIDE EFFECTS:
            ;2710 ;
            ;2711 ; D & Q Registers destroyed
            ;2712 ;--
            ;2713 DISABLE.CACHE:
U 11FC, 0818,0038,4580,0800,0000,11FD  ;2714  d_k[.8000]  ; ... and seed constant
U 11FD, 0500,003C,0180,0800,0000,11FE  ;2715  d_d.left  ; Generate final constant
U 11FE, 0819,0030,4580,0800,0000,11FF  ;2716  d_d.or.k[.8000]  ; ... = 00018000
U 11FF, 0000,003E,7580,3C00,0000,0001  ;2717  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2718  .PAGE
            ;2719  .TOC "   ENABLE CACHE"
            ;2720 ;++
            ;2721 ; FUNCTIONAL DESCRIPTION:
            ;2722 ;
            ;2723 ; This routine enables cache group 0 by clearing the force
            ;2724 ; miss bit in the maintenance register.
            ;2725 ;
            ;2726 ; CALLING CONSTRAINT:
            ;2727 ;
            ;2728 ; =0
            ;2729 ;
            ;2730 ; SIDE EFFECTS:
            ;2731 ;
            ;2732 ; D, Q AND SC Registers destroyed
            ;2733 ;--
            ;2734 ENABLE.CACHE:
U 1200, 0000.003C.75F0.2C00.0000.1201  ;2735  q_id[maint]  ; Get current maintenance register
U 1201, 0000.003C.6580.0800.0084.7202  ;2736  sc_k[.10]  ; Setup to clear bit 16
U 1202, 0801.2034.0180.0800.0000.1203  ;2737  d_q.and.mask  ; Clear bit 16
U 1203, 0000.003E.7580.3C00.0000.0001  ;2738  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2739  .PAGE
            ;2740  .TOC "   WAIT LOOP ROUTINE"
            ;2741 ;**
            ;2742 ; FUNCTIONAL DESCRIPTION:
            ;2743 ;
            ;2744 ; As the name implies,  this subroutine is used to set up a wait
            ;2745 ; loop for a specified  number of cycles.  This may be necessary
            ;2746 ; when the micro-program has to wait for another event to finish.
            ;2747 ; When the subroutine is entered Register D should be holding the
            ;2748 ; desired numbered of cycles.
            ;2749 ;
            ;2750 ; CALLING CONSTRAINT:
            ;2751 ;
            ;2752 ; =0
            ;2753 ;
            ;2754 ; INPUTS:
            ;2755 ;
            ;2756 ; d - Contains number of cycles to wait
            ;2757 ; alu.z condition code must not be set
            ;2758 ;
            ;2759 ; SIDE EFFECTS:
            ;2760 ;
            ;2761 ; d is destroyed
            ;2762 ;--
            ;2763 WAIT.LOOP:
U 1204, 0018,0038,6180,0800,0010,10D2  ;2764  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2765      ; ...is not set
            ;2766 =0
            ;2767 W.LOOP:
            ;2768  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10D2, 0819,0100,0580,0800,0010,10D2  ;2769       j/W.LOOP
U 10D3, 0000,003E,0180,0800,0000,0001  ;2770  return1    ; exit
            ;2771
```

```
            ;2772  .PAGE
            ;2773  .TOC "   CHECK FOR INTERRUPT ROUTINE"
            ;2774 ;++
            ;2775 ; FUNCTIONAL DESCRIPTION:
            ;2776 ;
            ;2777 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2778 ; If an interrupt is active, the following registers are setup:
            ;2779 ;  RC[8] - Gets the VECTOR register
            ;2780 ;  RC[7] - Gets the SBI ERROR register
            ;2781 ;  RC[6] - Gets the FAULT register
            ;2782 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2783 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2784 ;
            ;2785 ; CALLING CONSTRAINT:
            ;2786 ;
            ;2787 ; =00
            ;2788 ;
            ;2789 ; SIDE EFFECTS:
            ;2790 ;
            ;2791 ; D & Q are destroyed
            ;2792 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2793 ;--
            ;2794 CHECK_INTERRUPT:
            ;2795   ;
            ;2796   ; First, enable the fault and RDS/CRD interrupts
            ;2797   ;
U 1205, 0818,0038,4580,0800,0000,1206  ;2798  d_k[.8000]  ; Get data to set interrupt enable
            ;2799  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1206, 0500,003C,6580,3C00,0000,1207  ;2800      d_d.left
U 1207, 0100,003C,0180,0800,0000,1208  ;2801  d_d.left2  ; D = 40000
U 1208, 0000,003C,6D80,3C00,0000,1209  ;2802  id[fault]_d  ; Set fault interrupt enable
            ;2803  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1209, 0F03,003C,0180,09A8,4000,120A  ;2804      rc[5]_0  ; and clear interrupt occurred flag
U 120A, 0000,003C,3D80,3C00,0000,120B  ;2805  id[psl]_d  ; Clear the PSL
U 120B, 0000,003C,6580,3C00,0000,120C  ;2806  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 120C, 0000,003C,6D80,3C00,0000,120D  ;2807  id[fault]_d  ; Clear FAULT Interrupt Enable
U 120D, 0000,0E3C,0180,0800,0000,1004  ;2808  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2809 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2810  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2811  return1   ; No interrupt
            ;2812 =111
            ;2813 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,120E  ;2814  q_id[vector]  ; Get ID Vector Register
U 120E, 0001,203C,65F0,2DC0,0000,120F  ;2815  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 120F, 0001,203C,6DF0,2DB8,0000,1210  ;2816  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 1210, 0001,203C,0180,09B0,0000,1211  ;2817  rc[6]_q   ; Save falut reg
U 1211, 0018,003A,0580,09A8,0000,0002  ;2818  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2819  .PAGE
            ;2820  .TOC "   CHECK UTRAP"
            ;2821 ;++
            ;2822 ;FUNCTIONAL DESCRIPTION:
            ;2823 ;
            ;2824 ; This subroutine is used to check wether a Utrap occured.
            ;2825 ; It takes the contents of the Save Utrap flags register
            ;2826 ; R[0E], and compares them to the contrnts of the Utrap
            ;2827 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2828 ; according to the results of the test  (i.e., contents of
            ;2829 ; these registers are equal or not).
            ;2830 ; CALLING CONSTRAINT:
            ;2831 ;
            ;2832 ; =00
            ;2833 ;
            ;2834 ; INPUTS:
            ;2835 ;
            ;2836 ; R[0E]- Saved Utrap Mask
            ;2837 ; R[0F]- Expected Utrap Mask
            ;2838 ;
            ;2839 ;--
            ;2840 CHECK_UTRAP:
U 1212. 0800.003C.0180.0A70.0000.1213  ;2841  D_R[0E]    ; Reg D is loaded with the trap mask
U 1213. 0001.003C.0180.09C0.0000.1214  ;2842  RC[08]_D   ; Save expected Utrap flag for error logout
U 1214. 0800.003C.0180.0A7B.0000.1215  ;2843  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1215. 0001.003C.0180.09B8.0000.1216  ;2844  RC[07]_D   ; Save in RC[07]
            ;2845  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1216. 000D.0020.0180.0A70.0010.1217  ;2846          CLK.UBCC
U 1217. 0003.013C.0180.0AF8.0000.10D4  ;2847  Z?,R[0F]_0   ; Branch if no traps
U 10D4. 0000.003E.0180.0800.0000.0002  ;2848 =0 RETURN2    ; Utrap occured
U 10D5. 0000.003E.0180.0800.0000.0001  ;2849  RETURN1    ; No Utrap  return
            ;2850 ;
```

```
            ;2851  .PAGE
            ;2852  .TOC "    CHECK SBI.ERR"
            ;2853 ;++
            ;2854 ; FUNCTIONAL DESCRIPTION:
            ;2855 ;
            ;2856 ; This subroutine can be used to see whether the SBI.ERR Register
            ;2857 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;2858 ; routine will have to pass to this subroutine in the SC register
            ;2859 ; the number of the error bit to checked.
            ;2860 ;
            ;2861 ; CALLING CONSTRAINT:
            ;2862 ;
            ;2863 ; =00
            ;2864 ;
            ;2865 ; INPUTS:
            ;2866 ;
            ;2867 ; SC with the number of the error bit to be checked
            ;2868 ;
            ;2869 ; OUTPUTS:
            ;2870 ;
            ;2871 ; RC[08]-Contains the SBI.ERROR Register
            ;2872 ;--
            ;2873 CHECK_SBI_ERR:
U 1218, 0000,003C,65F0,2C00,0000,1219  ;2874  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 1219, 0001,203C,0180,09C0,0000,121A  ;2875  RC[08]_Q  ; Save for error logout
            ;2876  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 121A, 0001,2024,0180,0800,0010,121B  ;2877   CLK.UBCC ; ...
U 121B, 0000,013C,0180,0800,0000,10D6  ;2878  Z?   ; Check condition codes
U 10D6, 0000,003E,0180,0800,0000,0002  ;2879 =0 RETURN2   ; Bit is set
U 10D7, 0000,003E,0180,0800,0000,0001  ;2880  RETURN1   ; Bit is not set
```

```
            ;2881   .PAGE
            ;2882   .TOC "    SET TRAP MASK"
            ;2883 ;**
            ;2884 ; FUNCTIONAL DESCRIPTION:
            ;2885 ;
            ;2886 ; This routine is used to set a bit in the Micro Trap Mask
            ;2887 ; R[0F].
            ;2888 ;
            ;2889 ; CALLING CONSTRAINT:
            ;2890 ;
            ;2891 ; =0
            ;2892 ;
            ;2893 ; INPUTS:
            ;2894 ;
            ;2895 ; SC - Contains bit number to set.
            ;2896 ;
            ;2897 ; OUTPUTS:
            ;2898 ;
            ;2899 ; rc[0E] - Contains the current trap mask
            ;2900 ;
            ;2901 ; SIDE EFFECTS:
            ;2902 ;
            ;2903 ; d regsiter is destroyed
            ;2904 ;--
            ;2905 SET_TRAP_MASK:
U 121C, 0800,003C,0180,0A78,0000,121D  ;2906  d_r[0f]
U 121D, 0801,001C,0180,0AF8,0000,121E  ;2907  r[0f]_d.ornot.mask,d_alu ; Set mask
U 121E, 0001,003E,0180,0AF0,0000,0001  ;2908  r[0E]_d,return1    ; Save mask and return
```

```
            ;2909  .PAGE
            ;2910  .TOC "   FIND MS780-E MEMORY"
            ;2911 ;++
            ;2912 ; FUNCTIONAL DESCRIPTION:
            ;2913 ;
            ;2914 ;          The diagnostic is designed to handle up to 4 (four)
            ;2915 ;     MS780-E/H's on the 780 SBI. This routine looks at all TR levels
            ;2916 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2917 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2918 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2919 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2920 ;     first MS780-E found will have its starting address set to 0
            ;2921 ;     (zero).
            ;2922 ;     All the other MS780-E/H's found will have their starting address
            ;2923 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2924 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2925 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2926 ;     current MS780-E unit under test.
            ;2927 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
            ;2928 ;     set to maximum.
            ;2929 ;
            ;2930 ; CALLING CONSTRAINT:
            ;2931 ;
            ;2932 ; =00
            ;2933 ;
            ;2934 ; OUTPUTS:
            ;2935 ;
            ;2936 ; rc[0e] - Contains address of Configuration Register
            ;2937 ; r[0]  - Contains TR level
            ;2938 ;
            ;2939 ; SIDE EFFECTS:
            ;2940 ;
            ;2941 ; D register is destroyed.
            ;2942 ;--
            ;2943 =0
            ;2944 FIND.MS780E:
U 10D8. 0000,003D,0180,0800,0000,10D0  ;2945  call[sbi.init]  ; Make sure SBI is enabled
U 10D9. 0003,003C,0180,0988,0000,121F  ;2946  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 121F. 0818,0038,1980,0800,0000,1220  ;2947  d_k[zero]  ; Clear MS780-E/H counter
U 1220. 0000,003C,F980,3C00,0000,1221  ;2948  id[3e]_d  ; ...
U 1221. 0018,0038,0580,0A80,0000,1222  ;2949  r[0]_k[.1]  ; Init TR counter
U 1222. 0F03,003C,0180,09F0,0000,10DA  ;2950  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2951      ; ...CNFG A Reg
            ;2952 =0
            ;2953 FIND.MEM.LOOP:
U 10DA. 0800,003D,0180,0A00,0000,1248  ;2954  d_r[0],call[generate.io]; Generate address of TR level
U 10DB. 0001,003C,0180,09F0,0200,10DC  ;2955  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10DC. 0000,003D,8980,0800,0084,721C  ;2956 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2957  d[long]_cache.p, ; Read the specified tr level
U 10DD. 0000,003C,0180,C970,0000,1223  ;2958   ic_rc[0e]  ; ...
U 1223. 0000,003C,0180,0A78,0010,1224  ;2959  alu_r[0f],clk.ubcc ; Check for no response
U 1224. 0001,013C,0180,0880,0082,10DE  ;2960  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10DE. 0000,003C,0180,0800,0000,1225  ;2961 =0 j/check.adpt.code ; Check for a memory
U 10DF. 0000,003C,0180,0800,0000,1244  ;2962  j/find.mem.lp1  ; Try next tr
            ;2963 CHECK.ADPT.CODE:
```

```
U 1225, 0000.003C.1D80.0800.1404.7226  ;2964  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 1226, 0000.163C.0180.0800.0000.1028  ;2965  state7-4?  ; Branch on the adapter type
U 1028, 0000.003C.8980.0800.0084.7227  ;2966 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1029, 0000.003C.8980.0800.0084.7228  ;2967  j/ms780.c,sc_k[.d] ; MS780-C
U 102A, 0000.003C.0180.0800.0000.1244  ;2968  j/find.mem.lp1  ; Not a memory
U 102B, 0000.003C.0180.0800.0000.1244  ;2969  j/find.mem.lp1  ; Not a memory
U 102C, 0000.003C.6580.0800.0084.722D  ;2970  j/ma780.max,sc_k[.10] ; MA780
U 102D, 0000.003C.0180.0800.0000.1244  ;2971  j/find.mem.lp1  ; Not a memory
U 102E, 0010.0038.0180.09F0.0000.1231  ;2972  rc[0e]_lc,j/found.ms780e ; MS780-E
U 102F, 0010.0038.0180.09F0.0000.1231  ;2973  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2974   ;
          ;2975   ; Found an MS780-A or -C. Set the starting address
          ;2976   ; to maximum.
          ;2977   ;
U 1227, 0818.0038.5D80.0800.0000.1229  ;2978 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1228, 0818.0038.0580.0800.0000.1229  ;2979 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
          ;2980 MS780.COMMON:
U 1229, 0819.0030.7180.0800.0000.122A  ;2981  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 122A, 0000.003C.01F8.0803.0080.D22B  ;2982  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 122B, 0D00.003C.0180.0800.0000.122C  ;2983  d_dal.sc  ; Put data in bits<29:14>
          ;2984  cache.p_d[long], ; Set the starting address to maximum
U 122C, 0000.003C.0180.A800.0000.1244  ;2985       j/find.mem.lp1 ; and continue TR scan
          ;2986   ;
          ;2987   ; Found an MA780. Set the starting address to maximum
          ;2988   ;
          ;2989 MA780.MAX:
U 122D, 0818.0038.39F8.0803.0000.122E  ;2990  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 122E, 0D00.003C.0180.0803.0000.122F  ;2991  d_dal.sc,va_va+4 ; Put in proper position
U 122F, 0000.003C.0180.0803.0000.1230  ;2992  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;2993  cache.p_d[long], ; Set starting address to maximum
U 1230, 0000.003C.0180.A800.0000.1244  ;2994       j/find.mem.lp1
          ;2995 ;
          ;2996 ; Found an MS780E
          ;2997 ;
          ;2998 FOUND.MS780E:
U 1231, 0000.003C.F9F0.2C00.0000.1232  ;2999  q_id[3e]  ; Set up to see how many MS780-E's
          ;3000  alu_q.xor.k[.3], ; Are there Maximum units?
U 1232, 0019.2020.0D80.0800.0010.1233  ;3001   clk.ubcc ; Check condition codes
U 1233, 0000.013C.0180.0800.0000.10E0  ;3002  z?    ; Are we at maximum units for system
U 10E0, 0000.003C.0180.0800.0000.1234  ;3003 =0 J/ms780e.tst  ; No go find how many units ther are
U 10E1, 0818.0038.C180.0800.0000.10E2  ;3004  d_k[.ffff]  ; Yes set address to maximum
U 10E2, 0000.003D.0180.0800.0000.124C  ;3005 =0 call[set.start.addr] ; .....
U 10E3, 0000.003C.0180.0800.0000.1244  ;3006  j/find.mem.lp1  ; Go check to see if we are done
          ;3007 ;
          ;3008 MS780E.TST:
          ;3009  alu_rc[01],  ; Check "Found MS780E Flag"
U 1234, 0010.0038.0180.0908.0010.1235  ;3010   clk.ubcc ; Check condition codes
U 1235, 0810.0138.0180.0970.0000.10E4  ;3011  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;3012 =0 j/not.first.ms780e, ; No
U 10E4, 0818.0038.C180.0800.0000.10E8  ;3013   d_k[.ffff] ; Set starting address
U 10E5, 0001.003C.0180.0988.0000.1236  ;3014  rc[01]_d  ; Yes put base address in rc[01]
U 1236, 0818.0038.7980.0800.0000.1237  ;3015  d_k[.30]  ; Set up SC to point to first unit
U 1237, 0019.0014.F580.0800.0082.1238  ;3016  alu_d+k[.a],sc_alu ; ...
U 1238, 0810.0038.0180.0908.0000.1239  ;3017  d_rc[01]  ; Put base address of unit in D
U 1239, 0000.003C.0180.3400.0000.123A  ;3018  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 123A, 0F00,003C,0180,0800,0000,10E6  ;3019  d_0   ; Prepare to set starting address to Zero
U 10E6, 0000,003D,0180,0800,0000,124C  ;3020 =0 call[set.start.addr] ; Go do it
          ;3021  j/find.mem.lp1.  ; Check to see if we are done
U 10E7, 0003,003C,0180,09E8,0000,1244  ;3022   rc[0d]_0 ; ....
          ;3023 =0
          ;3024 NOT.FIRST.MS780E:
U 10E8, 0000,003D,0180,0800,0000,124C  ;3025  call[set.start.addr] ; Go set starting address to maximum
U 10E9, 0000,003C,F9F0,2C00,0000,123B  ;3026  q_id[3e]  ; Get the number of MS780-Es found
U 123B, 0819,2014,0580,0800,0000,123C  ;3027  d_q+k[.1]  ; Increment this counter
U 123C, 0000,003C,F980,3C00,0000,123D  ;3028  id[3e]_d  ; Save the counter
U 123D, 0018,0038,11C0,0800,0000,123E  ;3029  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;3030     ; ...were the I/O base addresses will be stored
U 123E, 081D,2000,7980,0800,0000,123F  ;3031  d_q-d,k[.30]  ; ... adjust pointer
U 123F, 0819,0014,7980,0800,0000,1240  ;3032  d_d+k[.30]  ; Point the SC to the ID register
U 1240, 0000,003C,F580,0800,0000,1241  ;3033  k[.a]   ; ...to receive I/O base address
U 1241, 0019,0014,F580,0800,0082,1242  ;3034  alu_d+k[.a],sc_alu ; ...
U 1242, 0810,0038,0180,0970,0000,1243  ;3035  d_rc[0e]  ; Get base address to d
U 1243, 0000,003C,0180,3400,0000,1244  ;3036  id(sc)_d  ; Move base address to ID pointed to by SC
          ;3037   ;
          ;3038   ; Try next tr
          ;3039   ;
          ;3040 FIND.MEM.LP1:
U 1244, 0818,0014,0580,0A80,0000,1245  ;3041  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;3042  alu_d.and.k[.f],
U 1245, 0019,0034,6180,0800,0010,1246  ;3043      clk.ubcc  ; Check if all done
U 1246, 0000,013C,0180,0900,0000,10EA  ;3044  z?   ; Branch if yes
U 10EA, 0000,003C,0180,0800,0000,10DA  ;3045 =0 j/find.mem.loop  ; Try next TR
U 10EB, 0810,0038,0180,0908,0010,1247  ;3046  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1247, 0000,013C,0180,0800,0000,10EC  ;3047  z?   ; Check condition codes
U 10EC, 0001,003E,0180,09F0,0000,0002  ;3048 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10ED, 0000,003E,0180,0800,0000,0001  ;3049  return1   ; No MS780E found
          ;3050
```

```
            ;3051  .PAGE
            ;3052  .TOC "   GENERATE IO ADDRESS"
            ;3053 ;**
            ;3054 ; FUNCTIONAL DESCRIPTION:
            ;3055 ;
            ;3056 ; This routine is used to generate an SBI Configuration Register
            ;3057 ; address from a TR level.
            ;3058 ;
            ;3059 ; CALLING CONSTRAINT:
            ;3060 ;
            ;3061 ; =0
            ;3062 ;
            ;3063 ; INPUTS:
            ;3064 ;
            ;3065 ; D - Contains TR level
            ;3066 ;
            ;3067 ; OUTPUTS:
            ;3068 ;
            ;3069 ; D - Contains SBI IO address
            ;3070 ;--
            ;3071 GENERATE.IO:
U 1248, 0000.003C.89F8.0800.0084.7249  ;3072  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1249, 0D00.003C.8D80.0800.0084.724A  ;3073  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 124A, 0000.003C.0980.0800.0084.B24B  ;3074  sc_sc-k[.2]  ; insert bit 29
U 124B, 0801.001E.0180.0800.0000.0001  ;3075  d_d.ornot.mask,return1 ; and return
```

```
            ;3076  .PAGE
            ;3077  .TOC "   SET STARTING ADDRESS"
            ;3078 ;**
            ;3079 ; FUNCTIONAL DESCRIPTION:
            ;3080 ;
            ;3081 ; This routine is used to set the starting address of the
            ;3082 ; memory to the specified value.
            ;3083 ;
            ;3084 ; CALLING CONSTRAINT:
            ;3085 ;
            ;3086 ; =0
            ;3087 ;
            ;3088 ; INPUTS:
            ;3089 ;
            ;3090 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3091 ;     (B Register Image divided by 2**16)
            ;3092 ; rc[0e] - Contains Address of Register A
            ;3093 ;
            ;3094 ; OUTPUTS:
            ;3095 ;
            ;3096 ; d - Contains aligned starting address (B Register Image)
            ;3097 ;
            ;3098 ; SIDE EFFECTS:
            ;3099 ;
            ;3100 ; d,q,va, and sc are destroyed
            ;3101 ;--
            ;3102 SET.START.ADDR:
U 124C, 0010,0038,6580,0970,0284,724D  ;3103  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 124D, 0000,003C,01F8,0803,0000,124E  ;3104  va_va+4,q_0  ; Set address to Register B
            ;3105  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 124E, 0D00,003C,0980,0800,0084,B24F  ;3106      sc_sc-k[.2]  ; Setup to insert bit 14
U 124F, 0801,001C,0180,0800,0000,1250  ;3107  d_d.ornot.mask  ; Insert Write Enable bit
U 1250, 0000,003E,0180,A800,0000,0001  ;3108  cache.p_d[long],return1 ; Set the starting address and return
```

```
            ;3109  .PAGE
            ;3110  .TOC "   ENABLE NEXT MS780-E"
            ;3111 ;**
            ;3112 ; FUNCTIONAL DESCRIPTION:
            ;3113 ;
            ;3114 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3115 ;    They will be tested  separately, according to the TR level at which
            ;3116 ;    they are located,  starting with the one  at the lowest level first.
            ;3117 ;          When  a test has finished  with the first unit, it will have to call
            ;3118 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3119 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3120 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3121 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3122 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3123 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3124 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3125 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3126 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3127 ;
            ;3128 ; CALLING CONSTRAINT:
            ;3129 ;
            ;3130 ;  =00
            ;3131 ;
            ;3132 ;--
            ;3133 ENABLE.NEXT.MS780E:
U 1251, 0000.003C.F9F0.2C00.0000.1252  ;3134  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3135  ALU_Q,   ; Check to see if it is zero
U 1252, 0001.203C.0180.0800.0010.1253  ;3136      CLK.UBCC  ; Check Condition Codes
U 1253, 0000.013C.0180.0800.0000.10EE  ;3137  Z?   ; ...for zero
U 10EE, 0000.003C.0180.0800.0000.1254  ;3138 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10EF, 0000.003E.0180.0800.0000.0002  ;3139  RETURN2   ; Zero No more units to test
            ;3140 ENABLE.NEXT:
U 1254, 0818.0038.C180.0800.0000.10F0  ;3141  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10F0, 0000.003D.0180.0800.0000.124C  ;3142 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10F1, 0818.0038.7980.0800.0000.1255  ;3143  D_K[.30]  ; Prepare to point to the ID register holding
            ;3144      ; ...the I/O Base address of the next MS780-E
U 1255, 0819.0014.1180.0800.0000.1256  ;3145  D_D+K[.4]  ; ...
U 1256, 0000.003C.F580.0800.0000.1257  ;3146  K[.A]   ; ...
U 1257, 0819.0014.F580.0800.0000.1258  ;3147  D_D+K[.A]  ; ...
U 1258, 0000.003C.F9F0.2C00.0000.1259  ;3148  Q_ID[3E]  ; Get MS780-E Counter
            ;3149  D_D-Q,   ; D now holds the pointer to the ID register
U 1259, 081D.0000.0180.0800.0082.125A  ;3150    SC_ALU  ; ...
U 125A, 0000.003C.01F0.2400.0000.125B  ;3151  Q_ID(SC)  ; Q gets adrress of next MS780E
U 125B, 0001.203C.0180.09F0.0000.125C  ;3152  RC[0E]_Q  ; Save it also in RC[0E]
U 125C, 0F03.003C.0180.09E8.0000.10F2  ;3153  RC[0D]_0,D_0  ; Set starting address to zero
U 10F2, 0000.003D.0180.0800.0000.124C  ;3154 =0 CALL[SET.START.ADDR] ; ....
U 10F3, 0F00.003C.F9F0.2C00.0000.125D  ;3155  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 125D, 081D.2008.0180.0800.0000.125E  ;3156  D_Q-D-1   ; Subtract 1 from total
U 125E, 0000.003C.F980.3C00.0000.10F4  ;3157  ID[3E]_D  ; Adjust the pointer
U 10F4, 0000.003D.0180.0800.0000.11F7  ;3158 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10F5, 0000.003E.0180.0800.0000.0001  ;3159  RETURN1   ; Tell caller another unit was found
            ;3160
```

```
            ;3161   .PAGE
            ;3162   .TOC "   LOAD TEMPORARY REGISTERS"
            ;3163 ;**
            ;3164 ; FUNCTIONAL DESCRIPTION
            ;3165 ;
            ;3166 ; This subroutine will be used by tests that check the data
            ;3167 ; integrity of some registers or memory location. It loads
            ;3168 ; the ID Temporary registers, T1 through T7 with some data
            ;3169 ; found in cache, the address of which has been previously
            ;3170 ; set up (before the call is made) by the calling routine
            ;3171 ; in the VA register.
            ;3172 ;
            ;3173 ; CALLING CONSTRAINT:
            ;3174 ;
            ;3175 ; =0
            ;3176 ;
            ;3177 ;
            ;3178 ; SIDE EFFECTS:
            ;3179 ;
            ;3180 ; The SC register will be used in this case as a pointer to
            ;3181 ; the ID Temporary registers.
            ;3182 ;
            ;3183 ;--
            ;3184 LOAD.TEMP.REGS:
U 125F. 0018.0038.5D80.0A88.0000.10F6   ;3185  R[1]_K[.7]   ; Set up counter for 7 temporary registers
U 10F6. 0000.003D.0180.0800.0000.1200   ;3186 =0 CALL[ENABLE.CACHE] ; Enable the cache
U 10F7. 001B.0010.7980.0800.0082.1260   ;3187  SC_K[.30]+1   ; Use the SC register as a pointer to the
            ;3188      ; ... ID temporary registers
            ;3189 LOAD.TEMP.LOOP1:
U 1260. 0000.003C.0180.C800.0000.1261   ;3190  D[LONG]_CACHE.P   ; Read a longword from cache
U 1261. 0000.003C.0180.3400.0000.1262   ;3191  ID(SC)_D   ; Leave the data in Temporary
            ;3192      ; ...register pointed by the SC
U 1262. 0000.003C.0180.0800.0080.D263   ;3193  SC_SC+1    ; Point to next Temp.REG
U 1263. 0000.003C.0180.0803.0000.1264   ;3194  VA_VA+4    ; Point to next longword of data in cache
U 1264. 0800.003C.0180.0A08.0000.1265   ;3195  D_R[1]     ; Prepare to decrement the counter
U 1265. 0019.0000.0580.0A88.0010.1266   ;3196  R[1]_D-K[.1],CLK.UBCC ; Decrement the counter by 1
U 1266. 0000.013C.0180.0800.0000.10F8   ;3197  Z?    ; Is the counter zero ?
U 10F8. 0000.003C.0180.0800.0000.1260   ;3198 =0 J/LOAD.TEMP.LOOP1 ; Jump if not zero else..
U 10F9. 0000.003C.0180.0800.0000.10FA   ;3199  NOP
U 10FA. 0000.003D.0180.0800.0000.11FC   ;3200 =0 CALL[DISABLE.CACHE] ; Disable the cache
U 10FB. 0000.003E.0180.0800.0000.0001   ;3201  RETURN1    ; Exit
            ;3202
```

```
                ;3203  .PAGE
                ;3204  .TOC "   REGISTER DATA INTEGRITY CHECK"
                ;3205 ;**
                ;3206 ;
                ;3207 ; FUNCTIONAL DESCRIPTION:
                ;3208 ;
                ;3209 ;       This subroutine is used in  conjunction with  the  previously discussed
                ;3210 ;       routine to check the data integrity of some register or memory location.
                ;3211 ;       The data  loaded by  the the  previous routine  is  a  set  of patterns
                ;3212 ;       designed  to replace the floating one and zero patterns.  This specific
                ;3213 ;       subroutine,  takes  the  data  from  the  ID  Temporary  registers  and
                ;3214 ;       writes  it  to some memory location or register whose address is passed
                ;3215 ;       down  from  the calling routine in the D  register.  The  data  is then
                ;3216 ;       read  back  masked with some  desired  bit mask  in  register RC[3] and
                ;3217 ;       compared  to the data  that  was  written.   If  the received data does
                ;3218 ;       not equal  the expected  data, an error call is made.  Error calls will
                ;3219 ;       be  made  in  the  case of unexpected  uTraps or interrupts. (Note: The
                ;3220 ;       first  longword  of  the  data  stored  in   cache   should  be  a  bit
                ;3221 ;       mask,  which  will  be  found  in ID  Temporary  register 1 (T1).  This
                ;3222 ;       mask  may  be  necessary   when  the  test should  check  only  certain
                ;3223 ;       bits  of  the  register  or  memory  location. If all bits  are  to  be
                ;3224 ;       checked,  the   mask   should obviously   be: xFFFFFFFF.)   By  using
                ;3225 ;       subroutines  to  perform  these  functions, the diagnostic  tests  will
                ;3226 ;       be more readable along with the fact that they can be used by more than
                ;3227 ;       one test in the same overlay  (thus saving some WCS storage space).
                ;3228 ;
                ;3229 ; CALLING CONSTRAINT:
                ;3230 ;
                ;3231 ; =0
                ;3232 ;
                ;3233 ; INPUTS:
                ;3234 ;
                ;3235 ; OUTPUTS:
                ;3236 ;
                ;3237 ; SIDE EFFECTS:
                ;3238 ;
                ;3239 ;--
                ;3240 ;*****************************************************************
                ;3241 DATA.CHECK:
U 1267. 0001.003C.0180.09D0.0000.1268  ;3242  RC[0A]_D  ; Save the address for error logout
U 1268. 0001.003C.0180.0800.0200.1269  ;3243  VA_D   ; Get the address to the VA register
U 1269. 001B.0010.7980.0B00.0082.126A  ;3244  SC_K[.30]+1  ; Point to the first Temp Register
U 126A. 0000.003C.01F0.2400.0000.126B  ;3245  Q_ID(SC)  ; Get the bit mask and save ...
U 126B. 0001.203C.0180.0998.0000.126C  ;3246  RC[03]_Q  ; ..in RC 3
U 126C. 0000.003C.0180.0800.0080.D26D  ;3247  SC_SC+1   ; Point to the first pattern
U 126D. 0018.0038.D580.DA90.0000.10FC  ;3248  R[2]_K[.6]  ; Set up a counter for 6 Temporary Registers
                ;3249 =0
                ;3250 DATA.CHECK.LOOP1:
U 10FC. 0000.003D.0180.0800.0000.11D0  ;3251  ERLOOP   ; Set the error loop
U 10FD. 0000.003C.0180.0800.0000.10FE  ;3252  NOP
U 10FE. 0000.003D.0180.0800.0000.10D0  ;3253 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10FF. 0000.003C.0180.0800.0000.126E  ;3254  NOP
U 126E. 0000.003C.01F0.2400.0000.126F  ;3255  Q_ID(SC)  ; Q gets test data
U 126F. 0C00.003C.0180.0800.0000.1270  ;3256  D_Q   ; Get the Memory/Register data
U 1270. 0000.003C.0180.0918.0000.1271  ;3257  LC_RC[03]  ; Save the expected data for error logout...
```

```
U 1271, 0011,0034,0180,09E0,0000,1272  ;3258  RC[0C]_D.AND.LC  ; ...and Mask
U 1272, 0000,003C,0180,A800,0000,1030  ;3259  CACHE.P_D[LONG]  ; Write it to the location pointed to by the VA
U 1030, 0000,003D,0180,0800,0000,1205  ;3260 =00 CHECK.INTERRUPT  ; Any Interrupts occured ?
U 1031, 0000,003C,0180,0800,0000,110A  ;3261  J/DATA.CHECK.NOINTR1 ; Go to loop 1 if no interrupt else..
          ;3262 ;
          ;3263 ;////////////////////////////////////////////////////////////////////////////
          ;3264 ;
U 1032, 0000,003D,F580,0800,0084,70C2  ;3265  ERROR2[.A]  ; Report error interrupt occured on write
          ;3266 ;
          ;3267 ;////////////////////////////////////////////////////////////////////////////
          ;3268 ;
          ;3269 ; ERROR DESCRIPTION:
          ;3270 ;
          ;3271 ; DATA:  I/O BASE ADDRESS CURRENTLY TESTED OF MS780E
          ;3272 ;  NOT USED
          ;3273 ;  EXPECTED DATA
          ;3274 ;         NOT USED
          ;3275 ;              ADDRESS UNDER TEST
          ;3276 ;         NOT USED
          ;3277 ;         INTERRUPT VECTOR
          ;3278 ;         SBI.ERROR REGISTER
          ;3279 ;         SBI.FAULT REGISTER
          ;3280 ;             INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
          ;3281 ;
          ;3282 ;////////////////////////////////////////////////////////////////////////////
          ;3283 ;
U 1033, 0000,003C,0180,0800,0000,110A  ;3284  NOP
          ;3285 =0
          ;3286 DATA.CHECK.NOINTR1:
U 110A, 0000,003D,0180,0800,0000,11D0  ;3287  ERLOOP  ; Set loop on error
U 110B, 0810,0038,0180,0960,0000,1273  ;3288  D_RC[0C]  ; Get ready to write the data to the register
          ;3289      ; ...again
U 1273, 0000,003C,0180,A800,0000,1110  ;3290  CACHE.P_D[LONG]  ; Do the Write
U 1110, 0000,003D,0180,0800,0000,10D0  ;3291 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1111, 0000,003C,0180,0800,0000,1112  ;3292  NOP
U 1112, 0000,003D,8980,0800,0084,721C  ;3293 =0 SET.TRAP.MASK[.D] ; Enable Time-Out Utraps
U 1113, 0000,003C,0180,0800,0000,1114  ;3294  NOP
U 1114, 0000,003D,8580,0800,0084,721C  ;3295 =0 SET.TRAP.MASK[.C] ; Enabl RDS Utraps
U 1115, 0000,003C,0180,C800,0000,1274  ;3296  D[LONG]_CACHE.P  ; Read back the location
U 1274, 0000,003C,0180,0918,0000,1275  ;3297  LC_RC[03]  ; Get the expected data
U 1275, 0011,0034,0180,09D8,0000,1034  ;3298  RC[0B]_D.AND.LC  ; Saved the received data for error logout-
          ;3299      ; ...and Mask unwanted bits
U 1034, 0000,003D,0180,0800,0000,1212  ;3300 =00 CHECK.UTRAP  ; Any Utraps occured ?
U 1035, 0000,003C,0180,0800,0000,1038  ;3301  J/DATA.CHECK.NOTRAP ; Jump if not
          ;3302 ;
```

```
            ;3303  .PAGE
            ;3304 ;//////////////////////////////////////////////////////////////////////////////
            ;3305 ;
U 1036, 0000.003D.0180.0800.0084.70C2  ;3306  ERROR2[.8]  ; Report error a Utrap occured on a read
            ;3307 ;
            ;3308 ;//////////////////////////////////////////////////////////////////////////////
            ;3309 ;
            ;3310 ; ERROR DESCRIPTION:
            ;3311 ;
            ;3312 ; DATA: I/O BASE CURRENTLY TESTED MS780E
            ;3313 ;  NOT USED
            ;3314 ;  EXPECTED DATA
            ;3315 ;  RECEIVED DATA
            ;3316 ;  REGISTER ADDRESS UNDER TEST
            ;3317 ;  NOT USED
            ;3318 ;  EXPECTED UTRAP FLAG
            ;3319 ;  RECEIVED UTRAP FLAG
            ;3320 ;
U 1037, 0000.003C.0180.0800.0000.1038  ;3321  NOP
            ;3322 =00
            ;3323 DATA.CHECK.NOTRAP:
U 1038, 0000.003D.0180.0800.0000.1205  ;3324  CHECK.INTERRUPT  ; Any Interrupts ?
U 1039, 0000.003C.0180.0800.0000.1276  ;3325  J/DATA.CHECK.NOINTR2 ; No branch to loop2
```

```
            ;3326  .PAGE
            ;3327 ;
            ;3328 ;////////////////////////////////////////////////////////////////////////////////
            ;3329 ;
U 103A, 0000,003D,F580,0800,0084,70C2  ;3330  ERROR2[.A]  ; Report error Interrupt occured on the read
            ;3331 ;
            ;3332 ;////////////////////////////////////////////////////////////////////////////////
            ;3333 ;
            ;3334 ; ERROR LOGOUT:
            ;3335 ;
            ;3336 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3337 ;  NOT USED
            ;3338 ;  EXPECTED DATA
            ;3339 ;  RECEIVED DATA
            ;3340 ;  REGISTER ADDRESS UNDER TEST
            ;3341 ;  NOT USED
            ;3342 ;  INTERRUPT VECTOR
            ;3343 ;  SBI.ERROR REGISTER
            ;3344 ;  SBI.FAULT REGISTER
            ;3345 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;3346 ;
            ;3347 ;////////////////////////////////////////////////////////////////////////////////
            ;3348 ;
U 103B, 0000,003C,0180,0800,0000,1276  ;3349  NOP
            ;3350 DATA.CHECK.NOINTR2:
U 1276, 0810,0038,0180,0958,0000,1277  ;3351  D_RC[0B]  ; Get Received data
            ;3352  ALU_D.XOR.RC[0C], ; Compare received to the expected data
U 1277, 0011,0020,0180,0960,0010,1278  ;3353   CLK.UBCC ; ...
U 1278, 0000,013C,0180,0800,0000,1116  ;3354  Z?   ; Are they equal ?
```

```
            ;3355  .PAGE
            ;3356 ;
            ;3357 ;////////////////////////////////////////////////////////////////////////////////
            ;3358 ;
U 1116. 0000.003D.D580.0800.0084.70C2  ;3359 =0 ERROR2[.6]  ; Bad data returned from memory
            ;3360 ;////////////////////////////////////////////////////////////////////////////////
            ;3361 ;
            ;3362 ; ERROR LOGOUT:
            ;3363 ;
            ;3364 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3365 ;  NOT USED
            ;3366 ;  EXPECTED DATA
            ;3367 ;  RECEIVED DATA
            ;3368 ;  REGISTER ADDRESS UNDER TEST
            ;3369 ;  NOT USED
            ;3370 ;
            ;3371 ;
            ;3372 ;////////////////////////////////////////////////////////////////////////////////
            ;3373 ;
            ;3374
U 1117. 0000.003C.0180.0800.0080.D279  ;3375  SC_SC+1   ; Point to next test location
U 1279. 0800.003C.0180.0A10.0000.127A  ;3376  D_R[2]    ; Prepare to decrement counter
U 127A. 0019.0000.0580.0A90.0010.127B  ;3377  R[2]_D-K[.1],CLK.UBCC ; Decrement counter by 1
U 127B. 0000.013C.0180.0800.0000.1118  ;3378  Z?    ; Done yet?
U 1118. 0000.003C.0180.0800.0000.10FC  ;3379 =0 J/DATA.CHECK.LOOP1 ; No  do until counter is zero
U 1119. 0000.003E.0180.0800.0000.0001  ;3380  RETURN1   ; EXIT
            ;3381
            ;3382
            ;3383  .TOC "   GET MEMORY SIZE"
            ;3384 ;**************************************************************
            ;3385 ;++
            ;3386 ;
            ;3387 ; Functional Description:
            ;3388 ; ---------------------
            ;3389 ;
            ;3390 ; This subroutine can be used to get the size of the memory.
            ;3391 ; CNFG Reg A Bits <13:09> have the memory size in 1 Meg
            ;3392 ; increments.
            ;3393 ;
            ;3394 ; Calling Constraint: =0
            ;3395 ; ------------------
            ;3396 ;
            ;3397 ;
            ;3398 ; Inputs:
            ;3399 ; ------
            ;3400 ;
            ;3401 ; RC[0E] Must have the Address of CNFG Register A
            ;3402 ;
            ;3403 ; Outputs:
            ;3404 ; -------
            ;3405 ;
            ;3406 ; Register D will contain upon exit the size of the
            ;3407 ; memory aligned on Mega-byte boundary.
            ;3408 ;
            ;3409 ; Side Effects:
```

```
            ;3410 ; ------------
            ;3411 ;
            ;3412 ; The contents of the following registers will be lost:
            ;3413 ;
            ;3414 ; D, SC, Q
            ;3415 ;
            ;3416 ;--
            ;3417 ;*************************************************************
            ;3418 GET.MEMORY.SIZE:
U 127C, 0010,0038,0180,0970,0200,127D  ;3419  VA_RC[0E]    ; Point to CNFG Reg A
U 127D, 0000,003C,0180,C800,0000,127E  ;3420  D[LONG]_CACHE.P    ; Read the register
U 127E, 001B,0000,D980,0800,0082,127F  ;3421  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 127F, 0D00,003C,0180,0800,0000,1280  ;3422  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 1280, 0819,0034,5580,0800,0000,103C  ;3423  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3424 =00 CALL[64K.OR.256K],   ; Find out the size of the arrays
U 103C, 0000,003D,01E0,0800,0000,11E4  ;3425   Q_D    ; Save Memory size bits in Q
U 103D, 0000,003D,0D80,0800,0084,70C0  ;3426  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3427       ; ...arrays on the controller
            ;3428 ;
            ;3429 ; The duplication of the next two return status is necessary so the returning
            ;3430 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3431 ; modifications.
            ;3432 ;
U 103E, 0819,2014,0580,0800,0000,103F  ;3433  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3434       ; ...(found 1 Meg arrays)
U 103F, 0819,2014,0580,0800,0000,1281  ;3435  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3436       ; ...(found 4 Meg arrays)
U 1281, 0000,003C,21F8,0800,0084,7282  ;3437  SC_K[.14],Q_0    ; Prepare to shift to Megabyte position
            ;3438  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 1282, 0D00,003E,0180,0800,0000,0001  ;3439   RETURN1    ; Return with memory size in Register D
            ;3440
            ;3441
```

```
                ;3442  .PAGE "TEST 185 FORCE TIME-OUT TEST"
                ;3443 ;*****************************************************
                ;3444 ;***
                ;3445 ;
                ;3446 ; Functional Description:
                ;3447 ; ----------------------
                ;3448 ;
                ;3449 ;   This test checks bit 8 (force timeout) of ID-1d (Maintenance Register)
                ;3450 ;   that it functions properly.
                ;3451 ;
                ;3452 ;
                ;3453 ; Error Logout:
                ;3454 ; ------------
                ;3455 ;
                ;3456 ; See individual error reports.
                ;3457 ;
                ;3458 ;=====================================================
                ;3459 ;
                ;3460 ; Register Map:
                ;3461 ; ------------
                ;3462 ;
                ;3463 ;
                ;3464 ;       RC        Description:
                ;3465 ;       --        -----------
                ;3466 ;
                ;3467 ;        E        I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
                ;3468 ;
                ;3469 ;--
                ;3470 ;*****************************************************
                ;3471 ;
                ;3472 =0
U 111A. 0000.003D.0180.0800.0000.10A3   ;3473 T.185: NEWTST    ; INCREMENT TEST NUMBER
U 111B. 0000.003C.0180.0800.0000.1040   ;3474  NOP
U 1040. 0000.003D.0180.0800.0000.10D8   ;3475 =00 CALL[FIND.MS780E]  ; Check for MS780-E on the SBI
U 1041. 0000.003C.0180.0800.0000.1123   ;3476  J/T185.EXIT    ; NO MS780-E FOUND
U 1042. 0000.003D.0180.0800.0000.10DD   ;3477  CALL[SBI.INIT]   ; Initialize the SBI
U 1043. 0000.003C.0180.0800.0000.1044   ;3478  NOP
                ;3479
U 1044. 0000.003D.0180.0800.0000.1024   ;3480 =00 CALL[SELECT.UPPER]  ; Try to configure the Upper controller
U 1045. 0000.003C.0180.0800.0000.1048   ;3481  J/T185.SELECT.LOWER  ; Misconfigured try to select the lower
U 1046. 0000.003C.0180.0800.0000.104A   ;3482  J/START.TEST.185  ; Configured OK go start the test
                ;3483 =
                ;3484 =00
                ;3485 T185.SELECT.LOWER:
U 1048. 0000.003D.0180.0800.0000.1020   ;3486  CALL[SELECT.LOWER]  ; Try to configure the lower controller
```

```
ZZ-ESKAR-V22  TEST 185 FORCE TIME-OUT TEST                                                    SEQ 0186
ESKAR3F.MCR                    MICRO2  1P(00)    26-JUL-85  17:04:54                        Page   9

            ;3487  .PAGE
            ;3488 ;
            ;3489 ;////////////////////////////////////////////////////////////////////////////////
            ;3490 ;
U 1049. 0000.003D.0580.0800.0084.70C0  ;3491  ERROR1[.1]   ; Couldn't Configure controllers
            ;3492 ;
            ;3493 ;////////////////////////////////////////////////////////////////////////////////
            ;3494 ;
            ;3495 ; ERROR LOGOUT:
            ;3496 ;
            ;3497 ; DATA: I/O BASE ADDRESS OF MS780E UNDER TEST
            ;3498 ;
            ;3499 ;////////////////////////////////////////////////////////////////////////////////
            ;3500 ;
            ;3501 =0
            ;3502 START.TEST.185:
U 104A. 0000.003D.0180.0800.0000.11D0  ;3503  ERLOOP
U 104B. 0000.003C.0180.0800.0084.7283  ;3504  SC_K[.8]    ;SEE FORCE TIMEOUT CLEAR
U 1283. 0000.003C.75F0.2C00.0000.1284  ;3505  Q_ID[MAINT]
U 1284. 0001.2024.0180.0800.0010.1285  ;3506  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1285. 0F00.013C.0180.0800.0000.111C  ;3507  Z?,D_0     ;IS FORCE TIME OUT CLEAR
U 111C. 0000.003D.0180.0800.0000.11D1  ;3508 =0 ERROR1     ;FORCE TIME OUT BIT DID NOT
            ;3509       ; CLEAR
U 111D. 0801.001C.75F0.2C00.0000.1286  ;3510  D_D.ORNOT.MASK,Q_ID[MAINT] ;SET FORCE TIMEOUT BIT
U 1286. 081D.0030.0180.0800.0084.7287  ;3511  D_D[OR]Q,SC_K[.8]  ;SET FORCE TIMEOUT AND CACHE OFF
U 1287. 0000.003C.7580.3C00.0000.1288  ;3512  ID[MAINT]_D    ;WRITE TIMEOUT BIT
U 1288. 0000.003C.75F0.2C00.0000.1289  ;3513  Q_ID[MAINT]    ;DID IT SET??
U 1289. 0001.2024.0180.0800.0010.128A  ;3514  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 128A. 0000.013C.0180.0800.0000.111E  ;3515  Z?
U 111E. 0000.003C.0180.0800.0000.128B  ;3516 =0 J/SD0010    ;YES IT'S SET
U 111F. 0000.003D.0180.0800.0000.11D1  ;3517  ERROR1     ;FORCE TIMEOUT DID NOT SET
U 128B. 0018.0038.1980.0800.0200.128C  ;3518 SD0010: VA_K[ZERO]    ;DO A READ OPERATION AT A
U 128C. 0F00.003C.8980.0800.0084.728D  ;3519  SC_K[.D],D_0    ;  VALID ADDRESS
U 128D. 0801.001C.0180.0AF8.0000.128E  ;3520  R[0F]_ALU,D_D.ORNOT.MASK ;CATCH TIMEOUTS
U 128E. 0000.003C.0180.C800.0000.128F  ;3521  MCT/READ.P    ;READ FORCING A TIMEOUT
U 128F. 0000.003C.0180.0A78.0010.1290  ;3522  ALU_R[0F],CLK.UBCC  ;DID IT TIMEOUT ???
U 1290. 0000.013C.0180.0800.0000.1120  ;3523  Z?
U 1120. 0000.003D.0180.0800.0000.11D1  ;3524 =0 ERROR1     ;NO TIMEOUT OCCURED
U 1121. 001B.0000.0580.0800.0200.1291  ;3525  VA_ALU,ALU_0-K[.1]  ; CHANGE THE VA, TIMEOUT ADDRESS
U 1291. 0000.003C.0180.0800.0000.1292  ;3526  NOP     ; REGISTER SHOULD STAY LOCKED UP
U 1292. 0000.003C.69F0.2C00.0000.1293  ;3527  Q_ID[TIME.ADDR]    ; READ THE REGISTER
U 1293. 0001.203C.0180.0800.0010.1294  ;3528  ALU_Q.CLK.UBCC    ; SHOULD BE ZERO
U 1294. 0000.013C.0180.0800.0000.1122  ;3529  Z?
U 1122. 0000.003D.0180.0800.0000.11D1  ;3530 =0 ERROR1     ; TIMEOUT ADDR REG DID NOT LOCK
            ;3531 T185.EXIT:
U 1123. 0000.003C.0180.0800.0000.1124  ;3532  NOP     ;END TEST
            ;3533
```

```
;3534   .PAGE "TEST 186 CNFG REGISTER ACCESS"
;3535 ;**
;3536 ; FUNCTIONAL DESCRIPTION:
;3537 ; ---------------------
;3538 ;
;3539 ; This test verifies whether the registers residing on the SBI interface
;3540 ; can be properly accessed, and it checks the integrity of some of the
;3541 ; bit fields in the Configuration Registers A and B.
;3542 ; The test consists of the following two subtests:
;3543 ;
;3544 ;       SUBTEST 1 - SBI INTERFACE REGISTER ACCESS
;3545 ;
;3546 ; This subtest checks register access for Read/Write operations to all
;3547 ; registers residing on the SBI Interface: CNFG A, CNFG B, Data Latch 1
;3548 ; and Data Latch 2.  It is expected that for normal operation, no Micro-
;3549 ; Traps or Interrupts will occur.  (The test is not concerned with data
;3550 ; integrity of these registers, but just with their accessibility.)
;3551 ;
;3552 ; SUBTEST 2 - CONFIGURATION REGISTER DATA INTEGRITY
;3553 ;
;3554 ; This subtest will test all Read/Write bits in Configuration Registers A
;3555 ; and B for stuck or shorted bits, by using test patterns stored in cache
;3556 ; at address 8 hex for CNFG A and  24 hex for CNFG B which are as
;3557 ; follows:
;3558   ;
;3559   ;x 8
;3560   ;$ 0007,0000,0100,0000,0102,0000,0103,0000
;3561   ;$ 0105,0000,0106,0000,0100,0000,0F7F,0FF8
;3562   ;$ 4000,0000,4555,0550,4AAA,0AA8,4333,0330
;3563   ;$ 4F0F,0F0F,40FF,00FF
;3564 ;
;3565 ;      CNFG Reg A              CNFG Reg B
;3566 ;
;3567 ; Mask:  x07                   x0FF80F7F
;3568 ;
;3569 ;       x100                  x00004000
;3570 ;
;3571 ;       x102                  x05504555
;3572 ;
;3573 ;       x103                  x0AA84AAA
;3574 ;
;3575 ;       x105                  x03304333
;3576 ;
;3577 ;       x106                  x0F0F4F0F
;3578 ;
;3579 ;       x100                  x00FF40FF
;3580 ;
;3581 ; The following bits will be tested:
;3582 ;
;3583 ;
;3584 ;                Bits         Function
;3585 ;                ----         --------
;3586 ;
;3587 ;   CNFG A         08          INTLV MODE WRITE ENABLE (write only)
;3588 ;
```

```
;3589 ;                <02:00>       INTERLEAVE MODE
;3590 ;
;3591 ;
;3592 ;   CNFG B       <27:19>       START ADDRESS
;3593 ;
;3594 ;                  14         START ADDR WR. EN. (WRITE ONLY)
;3595 ;
;3596 ;                  11         FORCE D BUS PARITY ERROR
;3597 ;
;3598 ;                <10:08>       DIAGNOSTIC MODE SELECT
;3599 ;
;3600 ;                <06:00>       DIAGNOSTIC ECC
;3601 ;
;3602 ; Error Logout:
;3603 ; ------------
;3604 ;
;3605 ; See individual reports!
;3606 ;
;3607 ;
;3608 ; LOGIC DESCRIPTION:
;3609 ; -----------------
;3610 ;
;3611 ;  N/A
;3612 ;
;3613 ;
;3614 ; REGISTER MAP:
;3615 ; ------------
;3616 ; RA,RB DESCRIPTION
;3617 ; -- -- -----------
;3618 ; 0 USED AS A COUNTER
;3619 ;
;3620 ;*******************************************************************************
;3621 ;
;3622 ; RC DESCRIPTION
;3623 ; -- -----------
;3624 ; E I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
;3625 ; D CNFG C/D ADDRESS OF CURRENTLY TESTED MS780-E
;3626 ; C SAVED BASE ADDRESS
;3627 ;
;3628 ;--
;3629 ;*******************************************************************************
;3630 =0
U 1124, 0000,003D,0180,0800,0000,10A3  ;3631 TEST.1: NEWTST    ;
U 1125, 0003,003C,0180,09E0,0000,1050  ;3632  RC[0C]_0   ; Clear Address Pointer
U 1050, 0000,003D,0180,0800,0000,10D8  ;3633 =00 CALL[FIND.MS780E] ; Check for MS780-E's on SBI
```

```
            ;3634  .PAGE
            ;3635 ;
            ;3636 ;////////////////////////////////////////////////////////////////////////////////
            ;3637 ;
U 1051. 0000.003D.1980.0800.0084.70C0  ;3638  ERROR1[ZERO]  ; No MS780E found to test
            ;3639 ;////////////////////////////////////////////////////////////////////////////////
            ;3640 ;
            ;3641 ; ERROR LOGOUT:
            ;3642 ;
            ;3643 ; DATA: NO DATA LOGOUT
            ;3644 ;
            ;3645 ;////////////////////////////////////////////////////////////////////////////////
            ;3646 ;
U 1052. 0000.003C.0180.0800.0000.1126  ;3647  NOP
            ;3648 =
            ;3649 =0
            ;3650 RESTART.TEST.1:
U 1126. 0000.003D.0180.0800.0000.11D8  ;3651  SUBTEST   ; INCREMENT SUBTEST NUMBER
U 1127. 0810.0038.0180.0970.0000.1295  ;3652  D_RC[0E]  ; Get  base address of MS780-E Unit
U 1295. 0001.003C.0180.09E0.0000.1296  ;3653  RC[0C]_D  ; Save it
U 1296. 0018.0038.D580.0A80.0000.1128  ;3654  R[0]_K[.6]  ; Set up a counter for registers:CNFG A and
            ;3655     ; CNFG B, ERR C, and D, DATA LATCHES 1 and 2
            ;3656 =0
            ;3657 TEST1.LOOP1:
U 1128. 0000.003D.0180.0800.0000.11D0  ;3658  ERLOOP   ; Pass U address for "loop on error"
U 1129. 0000.003C.0180.0800.0000.112A  ;3659  NOP
" 112A. 0000.003D.0180.0800.0000.10D0  ;3660 =0 CALL[SBI.INIT]  ; Clear the SBI
U 112B. 0010.0038.0180.0960.0200.112C  ;3661  VA_RC[0C]  ; Point to register under test
U 112C. 0000.003D.8980.0800.0084.721C  ;3662 =0 SET.TRAP.MASK[.D] ; Set expected timeout
U 112D. 0000.003C.0180.C800.0000.1054  ;3663  D[LONG]_CACHE.P  ; Attempt to read CNFG reg
U 1054. 0000.003D.0180.0800.0000.1212  ;3664 =00 CHECK.UTRAP  ; See if any Utraps occured
U 1055. 0000.003C.0180.0800.0000.1060  ;3665  J/TEST1.S1.NOTMO ; Jump if no traps else
```

```
ZZ-ESKAR-V22  TEST 186 CNFG REGISTER ACCESS                                                    SEQ 0190
ESKAR3F.MCR                    MICRO2  1P(00)    26-JUL-85  17:04:54                          Page  10

            ;3666  .PAGE
            ;3667 ;////////////////////////////////////////////////////////////////////////////////
            ;3668 ;
U 1056, 0000,003D,0180,0800,0084,70C2  ;3669  ERROR2[.8]  ; Report error a Micro-trap on register read
            ;3670 ;
            ;3671 ;////////////////////////////////////////////////////////////////////////////////
            ;3672 ;
            ;3673 ; ERROR DESCRIPTION:
            ;3674 ;
            ;3675 ; DATA: I/O BASE CURRENTLY TESTED MS780E
            ;3676 ;  NOT USED
            ;3677 ;  REGISTERR ADDRESS UNDER TEST
            ;3678 ;  NOT USED
            ;3679 ;  NOT USED
            ;3680 ;  NOT USED
            ;3681 ;  EXPECTED UTRAP FLAG
            ;3682 ;  RECEIVED UTRAP FLAG
            ;3683 ;
            ;3684 ;////////////////////////////////////////////////////////////////////////////////
            ;3685 ;
U 1057, 0000,003C,0180,0800,0000,1060  ;3686  NOP
            ;3687 =00
            ;3688 TEST1.S1.NOTMO:
U 1060, 0000,003D,0180,0800,0000,1205  ;3689  CHECK.INTERRUPT  ; Check for interrupts
U 1061, 0000,003C,0180,0800,0000,112E  ;3690  J/TEST1.S1.NOINT ; Jump if no interrupts else
```

```
            ;3691   .PAGE
            ;3692  ;
            ;3693  ;////////////////////////////////////////////////////////////////////////////
            ;3694  ;
U 1062, 0000,003D,F580,0800,0084,70C2  ;3695  ERROR2[.A]  ; Interrupt occured on register read
            ;3696  ;
            ;3697  ;////////////////////////////////////////////////////////////////////////////
            ;3698  ;
            ;3699  ; ERROR LOGOUT:
            ;3700  ;
            ;3701  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3702  ;   NOT USED
            ;3703  ;   REGISTER ADDRESS UNDER TEST
            ;3704  ;   NOT USED
            ;3705  ;   NOT USED
            ;3706  ;   NOT USED
            ;3707  ;   INTERRUPT VECTOR
            ;3708  ;   SBI.ERROR REGISTER
            ;3709  ;   SBI.FAULT REGISTER
            ;3710  ;   INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;3711  ;
            ;3712  ;////////////////////////////////////////////////////////////////////////////
U 1063, 0000,003C,0180,0800,0000,112E  ;3713  NOP
            ;3714  =0
            ;3715  TEST1.S1.NOINT:
U 112E, 0000,003D,0180,0800,0000,11D0  ;3716  ERLOOP    ; Pass Vaddress for "loop on error"
U 112F, 0000,003C,0180,0800,0000,1130  ;3717  nop
U 1130, 0000,003D,0180,0800,0000,10D0  ;3718 =0 CALL[SBI.INIT]  ; Clear the SBI
U 1131, 0000,003C,0180,A800,0000,1297  ;3719  CACHE.P_D[LONG]  ; Attempt to write the CNFG register
U 1297, 0F00,003C,D980,0800,0084,7298  ;3720  SC_K[.9],D_0  ; Get number of cycles to wait
U 1298, 0801,001C,0180,0800,0000,1132  ;3721  D_D.ORNOT.MASK  ; ...
U 1132, 0000,003D,0180,0800,0000,1204  ;3722 =0 CALL[WAIT.LOOP]  ; Wait for SBI to finish write cycle
U 1133, 0000,003C,0180,0800,0000,1064  ;3723  nop
U 1064, 0000,003D,8580,0800,0084,7218  ;3724 =00 CHECK.SBI.ERR[.C] ; See if there was an SBI time-out
U 1065, 0810,0038,0180,0960,0000,1067  ;3725  J/T1.S1.NOERR,D_RC[0C] ; Jump if there wasn't and error else
```

```
            ;3726  .PAGE
            ;3727 ;
            ;3728 ;////////////////////////////////////////////////////////////////////////////
            ;3729 ;
U 1066, 0000,003D,5D80,0800,0084,70C2  ;3730  ERROR2[.7]  ; Register write timed out
            ;3731 ;
            ;3732 ;////////////////////////////////////////////////////////////////////////////
            ;3733 ;
            ;3734 ; ERROR LOGOUT:
            ;3735 ;
            ;3736 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3737 ;  NOT USED
            ;3738 ;  REGISTER ADDRESS UNDER TEST
            ;3739 ;  NOT USED
            ;3740 ;  NOT USED
            ;3741 ;  NOT USED
            ;3742 ;  SBI.ERROR REGISTER
            ;3743 ;
            ;3744 ;////////////////////////////////////////////////////////////////////////////
            ;3745 ;
            ;3746
            ;3747 T1.S1.NOERR:
U 1067, 0019,0014,1180,09E0,0000,1299  ;3748  RC[0C]_D+K[.4]  ; Point to next register on the SBI
U 1299, 0800,003C,0180,0A00,0000,129A  ;3749  D_R[0]   ; Set up the counter
U 129A, 0019,0000,0580,0A80,0010,129B  ;3750  R[0]_D-K[.1],CLK.UBCC ; Subtract 1 from counter
U 129B, 0000,013C,0180,0800,0000,1134  ;3751  Z?   ; Are we finished ?
U 1134, 0000,003C,0180,0800,0000,1128  ;3752 =0 J/TEST1.LOOP1  ; No go test next register
U 1135, 0000,003C,0180,0800,0000,1136  ;3753  NOP
            ;3754 ;////////////////////////////////////////////////////////////////////////////
            ;3755 ;
            ;3756 ; SUBTEST 2 - CONFIGURATION  REGISTER DATA INTEGRITY
            ;3757 ;
            ;3758 ;////////////////////////////////////////////////////////////////////////////
            ;3759 ;
U 1136, 0000,003D,0180,0800,0000,11D8  ;3760 =0 subtest   ; Start subtest 2
U 1137, 0018,0038,0180,0800,0200,1138  ;3761  VA_K[.8]  ; Set up address for data in cache for CNFG A
U 1138, 0000,003D,0180,0800,0000,125F  ;3762 =0 CALL[LOAD.TEMP.REGS] ; Get test patterns from cache for CNFG A
U 1139, 0810,0038,0180,0970,0000,113A  ;3763  D_RC[0E]  ; Point to CNFG register A
U 113A, 0000,003D,0180,0800,0000,1267  ;3764 =0 CALL[DATA.CHECK] ; Go check the data
U 113B, 0018,0038,E980,0800,0200,113C  ;3765  VA_K[.24]  ; Set up address for data in cache for  CNFG B
U 113C, 0000,003D,0180,0800,0000,125F  ;3766 =0 CALL[LOAD.TEMP.REGS] ; Get test patterns from cache for CNFG B
U 113D, 0810,0038,0180,0970,0000,129C  ;3767  D_RC[0E]  ; Point to CNFG Register B
U 129C, 0819,0014,1180,0800,0000,113E  ;3768  D_D+K[.4]  ; ...
U 113E, 0000,003D,0180,0800,0000,1267  ;3769 =0 CALL[DATA.CHECK] ; Go check the data
U 113F, 0000,003C,0180,0800,0000,1068  ;3770  NOP
U 1068, 0000,003D,0180,0800,0000,1251  ;3771 =00 CALL[ENABLE.NEXT.MS780E]; Any more MS780-E'S ?
U 1069, 0000,003C,0180,0800,0000,1126  ;3772  J/RESTART.TEST.1 ; Restart test if there is else
U 106A, 0000,003C,0180,0800,0000,1140  ;3773  NOP   ; Finished
            ;3774 =
```

```
;3775  .PAGE "TEST 187 DATA LATCHES 1 AND 2 TEST"
;3776 ;*************************************************************
;3777 ;+++
;3778 ;
;3779 ; Functional Description:
;3780 ; ----------------------
;3781 ;
;3782 ;     This test checks the DATA LATCH 1 and DATA LATCH 2 for stuck
;3783 ; or  shorted  bits  by  using the patterns detailed below.
;3784 ; This  verifies  the
;3785 ; integrity of the INFORMATION FIELD transceivers, the data path to
;3786 ; the latches,  and the D  BUS.  Failure  of  this  test,  if  the
;3787 ; previous test passes, will indicate a fault in one or both of the
;3788 ; latches, although a failure could result from faulty SBI  PARITY,
;3789 ; ID,  MASK,  TAG,  or CNFG/ROM ADDRESS DECODE logic.  Most of this
;3790 ;  logic will be tested more specifically in later tests.
;3791 ;
;3792 ; The test patterns used in  this  test  are  as  follows:
;3793 ;
;3794 ;
;3795 ;  AAAAAAAA
;3796 ;  55555555
;3797 ;  33333333
;3798 ;  0F0F0F0F
;3799 ;  00FF00FF
;3800 ;  0000FFFF
;3801 ;
;3802 ; and they will be stored in cache at address x40
;3803 ;
;3804 ;
;3805 ; Logic Description:
;3806 ; -----------------
;3807 ;   N/A
;3808 ;
;3809 ;
;3810 ; Sync Point Description:
;3811 ; ----------------------
;3812 ;   N/A
;3813 ;
;3814 ; Error Logout:
;3815 ; ------------
;3816 ;
;3817 ; See individual error logout description in subroutine
;3818 ; DATA.CHECK.
;3819 ;
;3820 ;=============================================================================
;3821 ;
;3822 ; CACHE DATA FOR THIS TEST:
;3823 ;
;3824  ;% 40
;3825  ;$ FFFF,FFFF,AAAA,AAAA,5555,5555,3333,3333
;3826  ;$ 0F0F,0F0F,00FF,00FF,FFFF,0000
;3827 ;
;3828 ;
;3829 ;=============================================================================
```

```
            ;3830 ;
            ;3831 ; Register Map:
            ;3832 ; ------------
            ;3833 ;
            ;3834 ;      RA,RB      Description:
            ;3835 ;      -----      -----------
            ;3836 ;
            ;3837 ; Not Used
            ;3838 ;
            ;3839 ;
            ;3840 ;       RC        Description:
            ;3841 ;       --        -----------
            ;3842 ;
            ;3843 ; A Address under test (2000x010 for Data Latch 1
            ;3844 ;         2000x014 for Data Latch 2)
            ;3845 ;
            ;3846 ; B Received Data from the Latches
            ;3847 ;
            ;3848 ; C Expected Data from the Latches
            ;3849 ;
            ;3850 ; D Not Used
            ;3851 ;
            ;3852 ; E I/O Base Address for MS780-E/H under test
            ;3853 ;
            ;3854 ;
            ;3855 ;--
            ;3856 ;*************************************************************
            ;3857 =0
U 1140, 0000,003D,0180,0800,0000,10A3  ;3858 TEST.2: NEWTST     ; Increment test number
U 1141, 0000,003C,0180,0800,0000,106C  ;3859  NOP
U 106C, 0000,003D,0180,0800,0000,10D8  ;3860 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI
U 106D, 0000,003C,0180,0800,0000,1072  ;3861  J/TEST.2.EXIT    ; None Found
U 106E, 0018,0038,3180,0800,0200,1142  ;3862  VA_K[.40]    ; Point to cache address of the test data
            ;3863 =
U 1142, 0000,003D,0180,0800,0000,125F  ;3864 =0 CALL[LOAD.TEMP.REGS]  ; Load temporary registers T1 through
            ;3865       ; ...T7 with test data
            ;3866 RESTART.TEST.2:     ; Entry point for if more than 1
            ;3867       ; ...MS780-E/H on the SBI
U 1143, 0010,0038,01C0,0970,0000,129D  ;3868  Q_RC[0E]    ; Get MS780-E/H I/O base address
U 129D, 0819,2014,6580,0800,0000,1144  ;3869  D_Q+K[.10]    ; Point to Data Latch 1 on the
            ;3870       ; ...SBI Interface
U 1144, 0000,003D,0180,0800,0000,1267  ;3871 =0 CALL[DATA.CHECK]  ; Call subroutine to run test patterns
            ;3872       ; ...trhough Data Latch 1
U 1145, 0010,0038,01C0,0970,0000,129E  ;3873  Q_RC[0E]    ; Get MS780-E/H I/O Base Address
U 129E, 0819,2014,2180,0800,0000,1146  ;3874  D_Q+K[.14]    ; Point to Data Latch 2 on the
            ;3875       ; ...SBI Interface
U 1146, 0000,003D,0180,0800,0000,1267  ;3876 =0 CALL[DATA.CHECK]  ; Call subroutine to run test patterns
            ;3877       ; ...through Data Latch 2
U 1147, 0000,003C,0180,0800,0000,1070  ;3878  NOP     ; End the constraint block
U 1070, 0000,003D,0180,0800,0000,1251  ;3879 =00 CALL[ENABLE.NEXT.MS780E] ; Any more MS780-E/H's on the SBI?
U 1071, 0000,003C,0180,0800,0000,1143  ;3880  J/RESTART.TEST.2  ; Found another MS780-E/H
            ;3881 TEST.2.EXIT:
U 1072, 0000,003C,0180,0800,0000,1148  ;3882  NOP     ; Finished TEST 2
            ;3883 =
            ;3884
```

```
;3885  .PAGE "TEST 188 STARTING ADDRESS TEST"
;3886 ;************************************************************
;3887 ;***
;3888 ;
;3889 ; Functional Description:
;3890 ; ----------------------
;3891 ;
;3892 ;
;3893 ;      This sequence of tests checks the address  limits  logic  on
;3894 ; the  SBI  Interface,  more  specifically that the memory responds
;3895 ; only to addresses in the correct range (i.e.,  START  ADDRESS  <=
;3896 ; ACCESSED ADDRESS <= START ADDRS + MEMORY SIZE).  Failure of these
;3897 ; tests will most likely indicate a failure  of  the  ADDRESS  LIMT
;3898 ; logic on the SBI Interface, although it is also dependent on most
;3899 ; of the SBI Interface logic, the CONTROLLER  GOT  IT  confirmation
;3900 ; and  upon  the ARRAY  SIZE logic  on the  Controller board.  Both
;3901 ; tests  run certain  patterns through  the Starting Address field,
;3902 ; CNFG Reg B <27:19>, and accesses NXM with a Read operation. It is
;3903 ; expected that a Time-Out uTrap will bbe generated.
;3904 ;
;3905 ;
;3906 ;
;3907 ; SUBTEST 1 - STARTING ADDRESS TEST 1 -
;3908 ;
;3909 ;
;3910 ;      This subtest sets the starting address of the  memory  under
;3911 ; test  to  different patterns, and references addresses out of the
;3912 ; memory range.  It is intended  to  verify  the  comparators  that
;3913 ; generate  the signal "ADR GREAT H" (and, obviously the correlated
;3914 ; logic).
;3915 ; The patterns used will  be  stored  in  cache  and  they  are  as
;3916 ; follows:
;3917 ;
;3918 ;
;3919 ;   Starting Address <31:16>
;3920 ;        ----------------------
;3921 ;                  0210
;3922 ;                  01C0
;3923 ;                  0360
;3924 ;                  0020
;3925 ;                  0030
;3926 ;                  0040
;3927 ;                  0040
;3928 ;                  0080
;3929 ;                  0080
;3930 ;                  0080
;3931 ;                  0100
;3932 ;                  0100
;3933 ;                  0200
;3934 ;                  0200
;3935 ;                  0200
;3936 ;                  0200
;3937 ;                  0400
;3938 ;                  0400
;3939 ;                  0400
```

```
;3940 ;                    0400
;3941 ;                    0400
;3942 ;                    0800
;3943 ;                    0800
;3944 ;                    0800
;3945 ;                    0800
;3946 ;                    1000
;3947 ;                    1000
;3948 ;                    1000
;3949 ;                    1000
;3950 ;                    1900
;3951 ;
;3952 ; The last address + 1, determined by the memory size, will be added to
;3953 ; the starting address.
;3954 ;
;3955 ; SUBTEST 2 - STARTING ADDRESS TEST 2 -
;3956 ;
;3957 ;      This subtest is  similar  to  the  previous  one,  with  the
;3958 ; exception  that  the starting address to which the memory will be
;3959 ; set is calculated with the following formula:
;3960 ;
;3961 ;   Starting Address = Pattern from Table - Memory Size - 1
;3962 ;
;3963 ; This test is meant to check the  comparators  that  generate  the
;3964 ; signal "ADR LESS H".
;3965 ; As before the test patterns will be stored in cache and they  are
;3966 ; as follows:
;3967 ;
;3968 ;   Starting Address <31:16>          Referenced Address <31:16>
;3969 ;   ------------------------          --------------------------
;3970 ;                    02F0                              02F0
;3971 ;                    0010                              0020
;3972 ;                    0010                              0080
;3973 ;                    0010                              0040
;3974 ;                    0000                              0010
;3975 ;                    0020                              0040
;3976 ;                    0020                              0080
;3977 ;                    0040                              0080
;3978 ;
;3979 ;
;3980 ; Logic Description:
;3981 ; ----------------
;3982 ;   N/A
;3983 ;
;3984 ;
;3985 ; Sync Point Description:
;3986 ; ---------------------
;3987 ;   N/A
;3988 ;
;3989 ;=============================================================
;3990 ;
;3991 ; SUBTEST 1 CACHE DATA
;3992 ; -------------------
;3993 ;
;3994  ;% 80 ; Starting addresses patterns
```

```
            ;3995    ; All referenced address have a TAG of 0
            ;3996   ;$ 0210,01C0,0360,0020,0030,0040,0040,0080,0080,0080
            ;3997   ;$ 0100,0100,0200,0200,0200,0200,0400,0400,0400,0400,0400
            ;3998   ;$ 0800,0800,0800,0800,1000,1000,1000,1000,1900
            ;3999
            ;4000 ;
            ;4001 ; SUBTEST 2 CACHE DATA
            ;4002 ; --------------------
            ;4003 ;
            ;4004   ;% 100 ; Starting addresses patterns
            ;4005   ;$ 02F0,0010,0010,0010,0000,0020,0020,0040
            ;4006   ;
            ;4007   ;% 130 ; Referenced address patterns
            ;4008   ;$ 02F0,0020,0080,0040,0010,0040,0080,0080
            ;4009 ;
            ;4010 ;================================================================================
            ;4011 ;
            ;4012 ; Register Map:
            ;4013 ; ------------
            ;4014 ;
            ;4015 ;      RA,RB      Description:
            ;4016 ;      -----      -----------
            ;4017 ;
            ;4018 ;        0        CACHE ADDRESS OF "STARTING ADDRESS" PATTERNS
            ;4019 ;
            ;4020 ;        1        CACHE ADDRESS OF "REFERENCED ADDRESS" PATTERNS
            ;4021 ;
            ;4022 ;
            ;4023 ;       RC        Description:
            ;4024 ;       --        -----------
            ;4025 ;
            ;4026 ; A "MEMORY SIZE" BYTE ALIGNED
            ;4027 ;
            ;4028 ; B "REFERENCED ADDRESS" PATTERN
            ;4029 ;
            ;4030 ; C "STARTING ADDRESS" PATTERN
            ;4031 ;
            ;4032 ; D NOT USED
            ;4033 ;
            ;4034 ;        E        I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4035 ;
            ;4036 ;
            ;4037 ;--
            ;4038 ;*************************************************************
            ;4039 =0
U 1148.  0000,003D,0180,0800,0000,10A3   ;4040 TEST.4: NEWTST
U 1149.  0000,003C,0180,0800,0000,1074   ;4041  NOP
U 1074.  0000,003D,0180,0800,0000,10D8   ;4042 =00 CALL[FIND.MS780E] ; Find all MS780-E/Hs on the SBI
            ;4043      ; ...(Implicit enable of SBI)
U 1075.  0000,003C,0180,0800,0000,12C3   ;4044  j/TEST.4.EXIT  ; No memories
U 1076.  0000,003C,0180,0800,0000,1078   ;4045  nop   ;
            ;4046 =
            ;4047 =00
            ;4048 RESTART.TEST.4:    ; Entry point for Second Controller or
            ;4049      ; ...Next MS780-E/H
```

```
U 1078, 0000.003D.0180.0800.0000.10C6  ;4050  call[START.TEST] ; Configure Controllers
U 1079, 0000.003C.0180.0800.0000.12C3  ;4051  j/TEST.4.EXIT  ; No more Controllers or MS780-E/Hs
U 107A, 0000.003C.0180.0800.0000.114A  ;4052  nop
          ;4053 =
          ;4054 ;
          ;4055 ;=============================================================================
          ;4056 ;
          ;4057 ;  SUBTEST 1
          ;4058 ;
U 114A, 0000.003D.0180.0800.0000.11D8  ;4059 =0 subtest   ; Start SUBTEST 1
U 114B, 0018.0038.4180.0A80.0000.129F  ;4060  r[0]_k[.80]  ; Set cache address of "Starting Addresses"
          ;4061      ; ...Patterns
U 129F, 0018.0038.8D80.09C8.0000.114C  ;4062  rc[09]_k[.1f]  ; Set up Pattern counter for d31 patterns
          ;4063
U 114C, 0000.003D.0180.0800.0000.127C  ;4064 =0 CALL[GET.MEMORY.SIZE] ; Determine memory size based on Config Reg. B
U 114D, 0001.003C.0180.09D0.0000.114E  ;4065  rc[0A]_d  ; Routine GET.MEMORY.SIZE returns memory size
          ;4066      ; in the D register
U 114E, 0818.0039.0D80.0800.0000.11F0  ;4067 =0 set.diag.mode[.3] ; Set ECC bypass mode
U 114F, 0000.003C.0180.0800.0000.1150  ;4068  nop   ;
          ;4069 =0
          ;4070 TEST.4.LOOP1:
U 1150, 0000.003D.0180.0800.0000.1200  ;4071  call[enable.cache] ; Enable cache group 0
U 1151, 0F00.003C.0180.0A00.0200.12A0  ;4072  va_r[0],d_0  ; Get starting address from cache
U 12A0, 0000.403C.0180.C800.0000.12A1  ;4073  d[word]_cache.p  ; ...
U 12A1, 0019.0034.C180.09E0.0000.12A2  ;4074  rc[0C]_d.and.k[.FFFF] ;
U 12A2, 0010.0038.01C0.0950.0000.12A3  ;4075  q_rc[0A]  ; Get last address+1 based on memory size
U 12A3, 0000.003C.6580.0800.0084.72A4  ;4076  sc_k[.10]  ; Set-up to shift starting address left by 16
U 12A4, 0D00.003C.0180.0800.0000.12A5  ;4077  d_dal.sc  ; Shift starting address left by 16
U 12A5, 001D.0014.0180.09D8.0000.12A6  ;4078  rc[0B]_d+q  ; Add Starting Address to Last Address + 1
U 12A6, 0810.0038.0180.0960.0000.1152  ;4079  d_rc[0C]  ; Get starting address
          ;4080 =0 call[set.start.addr], ; Set the starting address
U 1152, 0600.003D.0180.0800.0000.124C  ;4081      d_d.right  ;
U 1153, 0000.003C.0180.0800.0000.1154  ;4082  nop
U 1154, 0000.003D.0180.0800.0000.11FC  ;4083 =0 call[disable.cache] ; Disable the cache
U 1155, 0000.003C.0180.0800.0000.1156  ;4084  nop
U 1156, 0000.003D.0180.0800.0000.11D0  ;4085 =0 ERLOOP   ; Set the error loop
U 1157, 0000.003C.0180.0800.0000.1158  ;4086  nop
U 1158, 0000.003D.0180.0800.0000.10D0  ;4087 =0 call[SBI.INIT]  ; Initialize the SBI
U 1159, 0000.003C.0180.0800.0000.115A  ;4088  nop
          ;4089 =0 set.trap.mask[.d],       ; Enable TmO uTraps
U 115A, 0010.0039.8980.0958.0284.721C  ;4090      va_alu,alu_rc[0B] ; ... and set Referenced address
U 115B, 0000.003C.0180.C800.0000.107C  ;4091  d[long]_cache.p  ; Execute a read from the address
U 107C, 0000.003D.0180.0800.0000.1212  ;4092 =00 CHECK.UTRAP  ; Check fo Time-Out uTrap
U 107D, 0000.003C.0180.0800.0000.115C  ;4093  j/TEST.4.NOTMO1  ; No Time-Out Trap Occured
U 107E, 0000.003C.0180.0800.0000.115D  ;4094  j/TEST.4.TMO1  ; Time-Out uTrap occured
U 107F, 0000.003C.0180.0800.0000.115C  ;4095  NOP
          ;4096
          ;4097 ;
          ;4098 ;/////////////////////////////////////////////////////////////////////////////
          ;4099 ;
          ;4100 =0
          ;4101 TEST.4.NOTMO1:
U 115C, 0000.003D.0180.0800.0084.70C2  ;4102  ERROR2[.8]  ; No Time-Out uTrap Occured
          ;4103      ; ...Flag the error
          ;4104 ;
```

```
            :4105 ;//////////////////////////////////////////////////////////////////////////////
            :4106 ;
            :4107 ; ERROR LOGOUT:
            :4108 ;
            :4109 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            :4110 ;  CNFG REGISTER C/D OF CONTROLLER USED
            :4111 ;  "STARTING ADDRESS" PATTERN WORD ALIGNED
            :4112 ;  "REFERENCED ADDRESS" PATTERN WORD ALIGNED
            :4113 ;  NOT USED
            :4114 ;  PATTERN COUNTER
            :4115 ;  BIT THAT SHOULD BE CLEAR IN THE RECEIVED MICRO-TRAP MASK
            :4116 ;  RECEIVED MICRO-TRAP MASK
            :4117 ;
            :4118 ;
            :4119 ;
            :4120 ;//////////////////////////////////////////////////////////////////////////////
            :4121 ;
            :4122
            :4123 TEST.4.TM01:
U 115D. 0800.003C.0180.0A00.0000.12A7  :4124  d_r[0]   ; Increment cache address pointer for
U 12A7. 0019.0014.0980.0A80.0000.12A8  :4125  r[0]_d+k[.2]  ; ..."Starting Address" Patterns
U 12A8. 0810.0038.0180.0948.0000.12A9  :4126  d_rc[09]  ; Decrement Pattern Counter
U 12A9. 0019.0000.0580.09C8.0010.12AB  :4127  rc[09]_d-k[.1],clk.ubcc ; ...and check if done
U 12AB. 0000.013C.0180.0800.0000.115E  :4128  z?   ; Branch if zero
U 115E. 0000.003C.0180.0800.0000.1150  :4129 =0 j/TEST.4.LOOP1  ; Try next pattern
U 115F. 0000.003C.0180.0800.0000.1160  :4130  NOP   ; Finished all Patterns
            :4131 ;
            :4132 ;==============================================================================
            :4133 ;
            :4134 ;  SUBTEST 2
            :4135 ;
            :4136 ;
U 1160. 0000.003D.0180.0800.0000.11D8  :4137 =0 subtest   ; Increment SUBTEST number
U 1161. 0838.0038.4180.0800.0000.12AC  :4138  d_alu.left,alu_k[.80] ;
U 12AC. 0001.003C.0180.0A80.0000.12AD  :4139  r[0]_d   ; Set Cache address of "Starting Addresses"
            :4140      ; ...Patterns
U 12AD. 0019.0030.7980.0A88.0000.12AE  :4141  r[1]_d.or.k[.30] ; Set Cache addresses of "Referenced Addresses"
            :4142      ; ...Patterns
U 12AE. 0018.0038.0180.09C8.0000.12AF  :4143  rc[09]_k[.8]  ; Set Pattern Counter for 8 patterns
U 12AF. 0010.0038.01F8.0970.0200.12B0  :4144  va_alu,alu_rc[0E],q_0 ; Get memory size bits
U 12B0. 0000.003C.0180.C800.0000.12B1  :4145  d[long]_cache.p  ; ...CNFG Reg A <14:09>
U 12B1. 001B.0008.1180.0800.0082.12B2  :4146  sc_0-k[.4]-1  ; Set the shift count (-5)
U 12B2. 0D00.003C.0180.0800.0000.12B3  :4147  d_dal.sc  ; Shift to bits <9:4>
U 12B3. 0018.0038.81C0.0800.0000.12B4  :4148  q_k[.3ff]  ; Generate mask for Mem-Size bits
U 12B4. 0019.2024.61C0.0800.0000.12B5  :4149  q_q.andnot.k[.f] ; eqls 3F0
U 12B5. 001D.0034.0180.09D0.0000.1162  :4150  rc[0A]_d.and.q  ; Mask and save memory size bits
            :4151 =0
            :4152 TEST.4.LOOP2:
U 1162. 0000.003D.0180.0800.0000.1200  :4153  call[enable.cache] ; Enable cache group 0
U 1163. 0F00.003C.0180.0A00.0200.12B6  :4154  va_r[0],d_0  ; Get starting address from cache
            :4155  d[word]_cache.p, ; ...
U 12B6. 0000.403C.0180.C950.0000.12B7  :4156      lc_rc[0A]  ; and latch Mem Size
U 12B7. 0011.0008.0180.09E0.0000.12B8  :4157  rc[0C]_alu,alu_d-lc-1 ; Adjust for this test
U 12B8. 0F00.003C.6580.0A08.0284.72B9  :4158  va_r[1],d_0,sc_k[.10] ; Get Referenced Address from cache
U 12B9. 0000.403C.01F8.C800.0000.12BA  :4159  d[word]_cache.p,q_0 ; ...
```

```
U 12BA. 0D00.003C.0180.0800.0000.12BB  ;4160  d_del.sc  ; ...Align it .
U 12BB. 0001.003C.0180.09D8.0000.12BC  ;4161  rc[0B]_d  ; ...Save it
U 12BC. 0810.0038.0180.0960.0000.1164  ;4162  d_rc[0C]  ; Get starting address
          ;4163 =0 call[SET.START.ADDR],
U 1164. 0600.003D.0180.0800.0000.124C  ;4164      d_d.right  ; Set the starting address
U 1165. 0000.003C.0180.0800.0000.1166  ;4165  nop
U 1166. 0000.003D.0180.0800.0000.11FC  ;4166 =0 call[disable.cache] ; Disable the cache
U 1167. 0000.003C.0180.0800.0000.1168  ;4167  nop
U 1168. 0000.003D.0180.0800.0000.11D0  ;4168 =0 ERLOOP   ; Set the error loop
U 1169. 0000.003C.0180.0800.0000.116A  ;4169  nop
U 116A. 0000.003D.0180.0800.0000.10D0  ;4170 =0 call[SBI.INIT]  ; Initialize the SBI
U 116B. 0000.003C.0180.0800.0000.116C  ;4171  nop
          ;4172 =0 set.trap.mask[.d], ; Set timeout trap flag and
U 116C. 0010.0039.8980.0958.0284.721C  ;4173      va_alu,alu_rc[0B] ; Set Referenced address
U 116D. 0000.003C.0180.C800.0000.1080  ;4174  d[long]_cache.p  ; Execute a read to the address
U 1080. 0000.003D.0180.0800.0000.1212  ;4175 =00 CHECK.UTRAP  ; Check for a Time-Out uTrap
U 1081. 0000.003C.0180.0800.0000.116E  ;4176  j/TEST.4.NOTMO2  ; No Time-Out uTrap Occurred
U 1082. 0000.003C.0180.0800.0000.116F  ;4177  j/TEST.4.TMO2  ; Time-Out Trap Occurred
U 1083. 0000.003C.0180.0800.0000.116E  ;4178  NOP
          ;4179
          ;4180 ;
          ;4181 ;////////////////////////////////////////////////////////////////////////////////
          ;4182 ;
          ;4183 =0
          ;4184 TEST.4.NOTMO2:
U 116E. 0000.003D.0180.0800.0084.70C2  ;4185  ERROR2[.8]  ; No Time-Out uTrap Occurred
          ;4186 ;
          ;4187 ;////////////////////////////////////////////////////////////////////////////////
          ;4188 ;
          ;4189 ; ERROR LOGOUT:
          ;4190 ;
          ;4191 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;4192 ;  CNFG C/D REGISTER ADDRESS OF CONTROLLER BEING USED
          ;4193 ;  "STARTING ADDRESS" PATTERN
          ;4194 ;  "REFERENCED ADDRESS" PATTERN
          ;4195 ;  MEMORY SIZE "BYTE ALIGNED"
          ;4196 ;  PATTERN COUNTER
          ;4197 ;  BIT EXPECTED TO BE CLEARED IN RECEIVED MICRO-TRAP MASK
          ;4198 ;  RECEIVED MICRO-TRAP MASK
          ;4199 ;
          ;4200 ;
          ;4201 ;
          ;4202 ;////////////////////////////////////////////////////////////////////////////////
          ;4203 ;
          ;4204 TEST.4.TMO2:
U 116F. 0800.003C.0180.0A00.0000.12BD  ;4205  d_r[0]   ; Increment cache address of "Starting
U 12BD. 0019.0014.0980.0A80.0000.12BE  ;4206  r[0]_d+k[.2]  ; ...Address" Patterns
U 12BE. 0800.003C.0180.0A08.0000.12BF  ;4207  d_r[1]   ; Increment cache address of "Referenced
U 12BF. 0019.0014.0980.0A88.0000.12C0  ;4208  r[1]_d+k[.2]  ; ...Address" Patterns
U 12C0. 0810.0038.0180.0948.0000.12C1  ;4209  d_rc[09]  ; Decrement Pattern Counter
U 12C1. 0019.0000.0580.09C8.0010.12C2  ;4210  rc[09]_d-k[.1],clk.ubcc ; ...and check if done
U 12C2. 0000.013C.0180.0800.0000.1170  ;4211  z?   ; Branch if done
U 1170. 0000.003C.0180.0800.0000.1162  ;4212 =0 j/TEST.4.LOOP2  ; Try next pattern. Not done yet.
U 1171. 0000.003C.0180.0800.0000.1078  ;4213  j/RESTART.TEST.4 ; Finished all patterns. Go see if
          ;4214     ; ...there are more Controllers or MS780-E/Hs
```

```
            ;4215 TEST.4.EXIT:
U 12C3, 0000,003C,0180,0800,0000,1172  ;4216  nop   ; Finished the test
            ;4217
```

```
;4218  .PAGE "TEST 189 CONTROLLER LATCHES"
;4219 ;*************************************************************
;4220 ;***
;4221 ;
;4222 ; Functional Description:
;4223 ; ---------------------
;4224 ;
;4225 ;      This test verifies the  integrity  of  the  Latches  on  the
;4226 ; Controller  board.   Data wrap-around on the controller board was
;4227 ; provided in Diagnostic Mode 1.   In this mode when performing  an
;4228 ; extended  write  to  memory, aside from the data being written on
;4229 ; the array, it will be latched on the controller board as follows:
;4230 ;
;4231 ;  -  Data passing through the LO Q Latch will be latched in
;4232 ;     the Rewrite Latch.
;4233 ;
;4234 ;  -  Data passing through the HI Q and HI X latches will be
;4235 ;           latched in the LO X latch.
;4236 ;
;4237 ; Extended reads will retreive the  data  from  these  latches  for
;4238 ; checking.    When  the read takes place, the path followed by the
;4239 ; data to the SBI is as follows:
;4240 ;
;4241 ;  -  Data latched in the Rewrite Latch will be passed through
;4242 ;     the LO Data OUT Latch
;4243 ;
;4244 ;  -  Data from the LO X Latch will be passed thorugh the
;4245 ;                HI Data OUT Latch.
;4246 ;
;4247 ; This wrap-around feature that was added, improves the ability  of
;4248 ; the diagnostic to detect errors on the Controller Board or on the
;4249 ; data paths between the SBI Interface and the Controller Boards.
;4250 ;
;4251 ; The test patterns used in  this  test  are  as  follows:
;4252 ;
;4253 ;
;4254 ;  AAAAAAAA
;4255 ;  55555555
;4256 ;  33333333
;4257 ;  0F0F0F0F
;4258 ;  00FF00FF
;4259 ;  0000FFFF
;4260 ;
;4261 ; and they will be stored in cache at address x40
;4262 ;
;4263 ; Logic Description:
;4264 ; -----------------
;4265 ;   N/A
;4266 ;
;4267 ; Error Logout:
;4268 ; ------------
;4269 ;
;4270 ; See individual error reports.
;4271 ;
;4272 ; Sync Point Description:
```

```
            ;4273 ; ----------------------
            ;4274 ;   N/A
            ;4275 ;
            ;4276 ;=============================================================
            ;4277 ;
            ;4278 ; Register Map:
            ;4279 ; ------------
            ;4280 ;
            ;4281 ;     RA,RB     Description:
            ;4282 ;     -----     -----------
            ;4283 ;
            ;4284 ;       0       REGISTER COUNTER
            ;4285 ;
            ;4286 ;
            ;4287 ;      RC       Description:
            ;4288 ;      --       -----------
            ;4289 ;
            ;4290 ; A RECEIVED DATA FROM CONTROLLER LOWER DATA PATHS
            ;4291 ;
            ;4292 ; B RECEIVED DATA FROM CONTROLLER UPPER DATA PATHS
            ;4293 ;
            ;4294 ; C EXPECTED DATA FROM LOWER AND UPPER
            ;4295 ;
            ;4296 ; D CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4297 ;
            ;4298 ;       E       I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4299 ;
            ;4300 ;
            ;4301 ;--
            ;4302 ;*************************************************************
            ;4303 =0
U 1172, 0000.003D,0180.0800,0000.10A3  ;4304 TEST.5: NEWTST    ; Flag beginning of new test
U 1173, 0000.003C,0180.0800,0000.1084  ;4305  NOP
U 1084, 0000.003D,0180.0800,0000.10D8  ;4306 =00 CALL[FIND.MS780E]  ; Any MS780-E/Hs on the SBI?
U 1085, 0000.003C,0180.0800,0000.12D5  ;4307  J/TEST.5.EXIT   ; No MS780-E/Hs found
U 1086, 0000.003C,0180.0800,0000.1088  ;4308  NOP
            ;4309 =
            ;4310 =00
            ;4311 RESTART.TEST.5:    ; Entry point for Second Controller
            ;4312       ; ...or next MS780-E
U 1088, 0000.003D,0180.0800,0000.10C6  ;4313  CALL[START.TEST]  ; Configure Controllers
U 1089, 0000.003C,0180.0800,0000.12D5  ;4314  J/TEST.5.EXIT   ; No more Controllers or MS780-E/Hs
U 108A, 0018.0038,3180.0800,0200.1174  ;4315  VA_K[.40]   ; Point to cache address of the test
            ;4316       ; ...data
            ;4317 =
U 1174, 0000.003D,0180.0800,0000.125F  ;4318 =0 CALL[LOAD.TEMP.REGS]  ; Get test data in Temporary registers
            ;4319       ; ...T1 through T7
U 1175, 0003.003C,0180.09C8,0000.12C4  ;4320  RC[09]_0   ; Location ACCESSED
U 12C4, 0838.0038,B980.0800,0000.12C5  ;4321  D_K[.19].LEFT.SI/ZERO  ; D is loaded with x32
U 12C5, 0001.003C,0180.0A88,0000.12C6  ;4322  R[1]_D    ; Save the Pointer to the Temp
            ;4323       ; ...Registers in R[1]
U 12C6, 0018.0038,D580.0A80,0000.1176  ;4324  R[0]_K[.6]   ; Set up Pattern Counter for 6 patterns
            ;4325 =0
            ;4326 TEST.5.LOOP:
U 1176, 0000.003D,0180.0800,0000.11D0  ;4327  ERLOOP    ; Set up the Error looping mechanism
```

```
U 1177. 0000.003C.0180.0800.0000.1178  ;4328  NOP
U 1178. 0000.003D.0180.0800.0000.10D0  ;4329 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1179. 0000.003C.0180.0800.0000.117A  ;4330  NOP
U 117A. 0818.0039.0580.0800.0000.11F0  ;4331 =0 SET.DIAG.MODE[.1]  ; Set Diagnostic Mode 1
U 117B. 0010.0038.0180.0948.0200.12C7  ;4332  VA_RC[09]   ; Point to location to be accessed
U 12C7. 0000.003C.0180.0A08.0082.12C8  ;4333  SC_R[1]    ; Point to Pattern in Temporary
        ;4334       ; ...Register to be used for test
U 12C8. 0000.003C.01F0.2400.0000.12C9  ;4335  Q_ID(SC)   ; Get the pattern in the Q reg
U 12C9. 0801.203C.0180.09E0.0000.12CA  ;4336  RC[0C]_Q.D_ALU   ; Save pattern fo error logout and
        ;4337       ; ...in the D Register to be written
U 12CA. 0000.003C.0180.A000.0000.108C  ;4338  MCT/EXTWRITE.P   ; Extended Write to the Controller
        ;4339       ; ..Data Paths
U 108C. 0000.003D.0180.0800.0000.1205  ;4340 =00 CHECK.INTERRUPT   ; Check for unexpected interrupts
U 108D. 0000.003C.0180.0800.0000.117C  ;4341  J/TEST.5.NOINTR1  ; OK. No interrupt Occurred
```

```
            ;4342  .PAGE
            ;4343 ;
            ;4344 ;////////////////////////////////////////////////////////////////////////////////
            ;4345 ;
U 108E, 0000,003D,F580,0800,0084,70C2  ;4346 =0 ERROR2[.A]    ; Unexpected Interrupt on Write
            ;4347       ; ...to Controller Data Paths
            ;4348 ;
            ;4349 ;////////////////////////////////////////////////////////////////////////////////
            ;4350 ;
            ;4351 ; ERROR LOGOUT:
            ;4352 ;
            ;4353 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4354 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4355 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4356 ;  NOT USED
            ;4357 ;  NOT USED
            ;4358 ;  NOT USED
            ;4359 ;  ID VECTOR REGISTER
            ;4360 ;  ID SBI.ERR REGISTER
            ;4361 ;  ID FAULT REGISTER
            ;4362 ;  INTERRUPT OCCURRED FLAG    (0 - NO INTERRUPT)
            ;4363 ;         (1 - INTERRUPT OCCURRED)
            ;4364 ;
            ;4365 ;
            ;4366 ;////////////////////////////////////////////////////////////////////////////////
            ;4367 ;
U 108F, 0000,003C,0180,0800,0000,117C  ;4368  NOP
            ;4369 =0
            ;4370 TEST.5.NOINTR1:
U 117C, 0000,003D,0180,0800,0000,11D0  ;4371  ERLOOP     ; Set the error looping mechanism
U 117D, 0810,0038,0180,0960,0000,12CB  ;4372  D_RC[0C]    ; Get data to perform another write
            ;4373       ; ...to the Controller Data Paths
U 12CB, 0000,003C,0180,A000,0000,117E  ;4374  MCT/EXTWRITE.P    ; Do the Extended write
U 117E, 0000,003D,8580,0800,0084,721C  ;4375 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 117F, 0000,003C,0180,0800,0000,1180  ;4376  NOP
U 1180, 0000,003D,8980,0800,0084,721C  ;4377 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 1181, 0000,003C,0180,C800,0000,1090  ;4378  D[LONG]_CACHE.P    ; Read the Lower Data Paths
            ;4379 =00 RC[0A]_D,    ; Save for error logout
U 1090, 0001,003D,0180,09D0,0000,1212  ;4380   CHECK.UTRAP  ; Check for occurrence of RDS
            ;4381       ; ...or Time-Out uTraps
U 1091, 0000,003C,0180,0800,0000,1094  ;4382  J/TEST.5.NOTRP1    ; OK. No uTrap on Read
```

```
            ;4383  .PAGE
            ;4384 ;
            ;4385 ;////////////////////////////////////////////////////////////////////////////////
            ;4386 ;
U 1092, 0000,003D,0180,0800,0084,70C2  ;4387 =0 ERROR2[.8]   ; Some uTrap occurred on Read from
            ;4388      ; ...COntroller Lower Data  Paths
            ;4389 ;
            ;4390 ;////////////////////////////////////////////////////////////////////////////////
            ;4391 ;
            ;4392 ; ERROR LOGOUT:
            ;4393 ;
            ;4394 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4395 ;  CNFG REGISTER C/D OF CONTROLLER CURRENTLY UNDER TEST
            ;4396 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4397 ;  NOT USED
            ;4398 ;  RECEIVED DATA FROM CONTROLLER LOWER DATA PATHS
            ;4399 ;  NOT USED
            ;4400 ;  BITS EXPECTED TO BE SET IN THE RECEIVED MICRO-TRAP MASK
            ;4401 ;  RECEIVED MICRO-TRAP MASK
            ;4402 ;
            ;4403 ;
            ;4404 ;////////////////////////////////////////////////////////////////////////////////
            ;4405 ;
U 1093, 0000,003C,0180,0800,0000,1094  ;4406  NOP
            ;4407 =00
            ;4408 TEST.5.NOTRP1:
U 1094, 0000,003D,0180,0800,0000,1205  ;4409  CHECK.INTERRUPT   ; Check for unexpected interrupts
U 1095, 0000,003C,0180,0800,0000,1097  ;4410  J/TEST.5.NOINTR2
            ;4411
            ;4412 ;
            ;4413 ;////////////////////////////////////////////////////////////////////////////////
            ;4414 ;
U 1096, 0000,003D,F580,0800,0084,70C2  ;4415 =0 ERROR2[.A]   ; Unexpected Interrupt on Read
            ;4416      ; ...from Controller Lower Data Paths
            ;4417 ;
            ;4418 ;////////////////////////////////////////////////////////////////////////////////
            ;4419 ;
            ;4420 ; ERROR LOGOUT:
            ;4421 ;
            ;4422 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4423 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4424 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4425 ;  NOT USED
            ;4426 ;  RECEIVED DATA FROM LOWER DATA PATHS
            ;4427 ;  NOT USED
            ;4428 ;  ID VECTOR REGISTER
            ;4429 ;  ID SBI.ERR REGISTER
            ;4430 ;  ID FAULT REGISTER
            ;4431 ;  INTERRUPT OCCURRED FLAG    (0 - NO INTERRUPT)
            ;4432 ;         (1 - INTERRUPT OCCURRED)
            ;4433 ;
            ;4434 ;
            ;4435 ;////////////////////////////////////////////////////////////////////////////////
            ;4436 ;
            ;4437 TEST.5.NOINTR2:
```

```
U 1097, 0810,0038,0180,0950,0000,12CC  ;4438  D_RC[0A]   ; Get Received data from Lower
          ;4439       ; ...Data Paths
          ;4440  ALU_D.XOR.RC[0C],  ; Check against expected data
U 12CC, 0011,0020,0180,0960,0010,12CD  ;4441   CLK.UBCC  ; ...
U 12CD, 0000,013C,0180,0800,0000,1182  ;4442  Z?
```

```
          ;4443  .PAGE
          ;4444 ;
          ;4445 ;////////////////////////////////////////////////////////////////////////////
          ;4446 ;
U 1182, 0000,003D,D580,0800,0084,70C2  ;4447 =0 ERROR2[.6]   ; Wrong Data returned from Controller
          ;4448      ; ...Lower Data Paths
          ;4449 ;
          ;4450 ;////////////////////////////////////////////////////////////////////////////
          ;4451 ;
          ;4452 ; ERROR LOGOUT:
          ;4453 ;
          ;4454 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;4455 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
          ;4456 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
          ;4457 ;  NOT USED
          ;4458 ;  RECEIVED DATA FROM CONTROLLER LOWER DATA PATHS
          ;4459 ;  NOT USED
          ;4460 ;
          ;4461 ;
          ;4462 ;////////////////////////////////////////////////////////////////////////////
          ;4463 ;
U 1183, 0000,003C,0180,0800,0000,1184  ;4464  NOP
U 1184, 0000,003D,0180,0800,0000,11D0  ;4465 =0 ERLOOP    ; Set the error looping mechanism
U 1185, 0810,0038,0180,0960,0000,12CE  ;4466  D_RC[0C]    ; Get Data to reinitialize the
          ;4467      ; ...Controller Data Path registers
U 12CE, 0000,003C,0180,A000,0000,1186  ;4468  MCT/EXTWRITE.P   ; Do the Extended Write
U 1186, 0000,003D,8580,0800,0084,721C  ;4469 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 1187, 0000,003C,0180,0800,0000,1188  ;4470  NOP
U 1188, 0000,003D,8980,0800,0084,721C  ;4471 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 1189, 0000,003C,0180,C800,0000,1098  ;4472  D[LONG]_CACHE.P   ; Read the Upper Data Paths
          ;4473 =00 RC[0B]_D,    ; Save for error logout
U 1098, 0001,003D,0180,09D8,0000,1212  ;4474   CHECK.UTRAP  ; Check for occurrence of RDS
          ;4475      ; ...or Time-Out uTraps
U 1099, 0000,003C,0180,0800,0000,109C  ;4476  J/TEST.5.NOTRP2   ; OK. No uTrap on Read
```

```
            ;4477  .PAGE
            ;4478 ;
            ;4479 ;////////////////////////////////////////////////////////////////////////////////
            ;4480 ;
U 109A, 0000.003D.0180.0800.0084.70C2  ;4481 =0 ERROR2[.8]   ; Some uTrap occurred on Read from
            ;4482       ; ...Controller Upper Data  Paths
            ;4483 ;
            ;4484 ;////////////////////////////////////////////////////////////////////////////////
            ;4485 ;
            ;4486 ; ERROR LOGOUT:
            ;4487 ;
            ;4488 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4489 ;  CNFG REGISTER C/D OF CONTROLLER CURRENTLY UNDER TEST
            ;4490 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4491 ;  RECEIVED DATA FROM CONTROLLER UPPER DATA PATHS
            ;4492 ;  RECEIVED DATA FROM CONTROLLER LOWER DATA PATHS
            ;4493 ;  NOT USED
            ;4494 ;  BITS EXPECTED TO BE SET IN THE RECEIVED MICRO-TRAP MASK
            ;4495 ;  RECEIVED MICRO-TRAP MASK
            ;4496 ;
            ;4497 ;
            ;4498 ;////////////////////////////////////////////////////////////////////////////////
            ;4499 ;
U 109B, 0000.003C.0180.0800.0000.109C  ;4500  NOP
            ;4501 =00
            ;4502 TEST.5.NOTRP2:
U 109C, 0000.003D.0180.0800.0000.1205  ;4503  CHECK.INTERRUPT   ; Check for unexpected interrupts
U 109D, 0000.003C.0180.0800.0000.109F  ;4504  J/TEST.5.NOINTR3  ; OK. No unexpected interrupts
            ;4505
            ;4506 ;
            ;4507 ;////////////////////////////////////////////////////////////////////////////////
            ;4508 ;
U 109E, 0000.003D.F580.0800.0084.70C2  ;4509 =0 ERROR2[.A]   ; Unexpected Interrupt on Read
            ;4510       ; ...from Controller Lower Data Paths
            ;4511 ;
            ;4512 ;////////////////////////////////////////////////////////////////////////////////
            ;4513 ;
            ;4514 ; ERROR LOGOUT:
            ;4515 ;
            ;4516 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4517 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4518 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4519 ;  RECEIVED DATA FROM UPPER DATA PATHS
            ;4520 ;  RECEIVED DATA FROM LOWER DATA PATHS
            ;4521 ;  NOT USED
            ;4522 ;  ID VECTOR REGISTER
            ;4523 ;  ID SBI.ERR REGISTER
            ;4524 ;  ID FAULT REGISTER
            ;4525 ;  INTERRUPT OCCURRED FLAG    (0 - NO INTERRUPT)
            ;4526 ;         (1 - INTERRUPT OCCURRED)
            ;4527 ;
            ;4528 ;
            ;4529 ;////////////////////////////////////////////////////////////////////////////////
            ;4530 ;
            ;4531 TEST.5.NOINTR3:
```

```
U 109F, 0810,0038,0180,0958,0000,12CF  ;4532  D_RC[0B]   ; Get Received data from Lower
          :4533       ; ...Data Paths
          :4534  ALU_D.XOR.RC[0C],  ; Check against expected data
U 12CF, 0011,0020,0180,0960,0010,12D0  ;4535   CLK.UBCC  ; ...
U 12D0, 0000,013C,0180,0800,0000,118A  ;4536  Z?
```

```
            ;4537  .PAGE
            ;4538 ;
            ;4539 ;////////////////////////////////////////////////////////////////////////////////
            ;4540 ;
U 118A. 0000.003D.D580.0800.0084.70C2  ;4541 =0 ERROR2[.6]   ; Wrong Data returned from Controller
            ;4542      ; ...Upper Data Paths
            ;4543 ;
            ;4544 ;////////////////////////////////////////////////////////////////////////////////
            ;4545 ;
            ;4546 ; ERROR LOGOUT:
            ;4547 ;
            ;4548 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4549 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4550 ;  EXPECTED DATA FROM LOWER AND UPPER DATA PATHS
            ;4551 ;  RECEIVED DATA FROM CONTROLLER UPPER DATA PATHS
            ;4552 ;  RECEIVED DATA FROM CONTROLLER LOWER DATA PATHS
            ;4553 ;  NOT USED
            ;4554 ;
            ;4555 ;
            ;4556 ;////////////////////////////////////////////////////////////////////////////////
            ;4557 ;
U 118B. 0800.003C.0180.0A08.0000.12D1  ;4558  D_R[1]     ; Get pointer to test patterns
U 12D1. 0019.0014.0580.0A88.0000.12D2  ;4559  R[1]_D+K[.1]   ; Increment the pointer
U 12D2. 0800.003C.0180.0A00.0000.12D3  ;4560  D_R[0]     ; Get loop Counter
            ;4561  R[0]_D-K[.1],   ; Decrement Loop Counter
U 12D3. 0019.0000.0580.0A80.0010.12D4  ;4562   CLK.UBCC  ; See if it is 0
U 12D4. 0000.013C.0180.0800.0000.118C  ;4563  Z?     ; ...
U 118C. 0000.003C.0180.0800.0000.1176  ;4564 =0 J/TEST.5.LOOP   ; Not finished all patterns yet
U 118D. 0000.003C.0180.0800.0000.1088  ;4565  J/RESTART.TEST.5  ; See if it is necessary to repeat
            ;4566      ; ...the test
            ;4567 TEST.5.EXIT:
U 12D5. 0000.003C.0180.0800.0000.118E  ;4568  NOP     ; Finished the test
            ;4569
```

```
;4570  .PAGE "TEST 18A BASIC MEMORY WRITE READ"
;4571 ;*************************************************************
;4572 ;+++
;4573 ;
;4574 ; Functional Description:
;4575 ; ---------------------
;4576 ;
;4577 ;      This test verifies the ability to  perform  EXTENDED  WRITES
;4578 ; and  EXTENDED  READS.  This test is neccesary at this point since
;4579 ; subsequent tests depend upon the ability of reading  and  writing
;4580 ; the arrays.  More extensive array testing will be performed after
;4581 ; checking the SBI Interface and Controller logic.
;4582 ;
;4583 ;      Locations 0 and 4 will be tested for Read/Write  access  and
;4584 ; for  data  integrity  (using the patterns in Appendix B) with the
;4585 ; ECC logic turned off and with it on.
;4586 ;
;4587 ;
;4588 ;
;4589 ; SUBTEST 1 - TEST WITH ECC OFF -
;4590 ;
;4591 ;
;4592 ; SUBTEST 2 - TEST WITH ECC ON -
;4593 ;
;4594 ;
;4595 ; Logic Description:
;4596 ; -----------------
;4597 ;
;4598 ;   N/A
;4599 ;
;4600 ; Error Logout:
;4601 ; ------------
;4602 ;
;4603 ; See individual error reports.
;4604 ;
;4605 ; Sync Point Description:
;4606 ; ----------------------
;4607 ;
;4608 ;   N/A
;4609 ;
;4610 ;=============================================================
;4611 ;
;4612 ; Register Map:
;4613 ; ------------
;4614 ;
;4615 ;     RA,RB     Description:
;4616 ;     -----     -----------
;4617 ;
;4618 ;       0       REGISTER COUNTER
;4619 ;
;4620 ; 1 TEMPORARY REGISTER POINTER "SAVE" LOCATION
;4621 ;
;4622 ; E SAVE uTRAP MASK
;4623 ;
;4624 ;       F       uTRAP MASK
```

```
            ;4625 ;
            ;4626 ;
            ;4627 ;       RC        Description:
            ;4628 ;       --        -----------
            ;4629 ;
            ;4630 ; 4 CODE FOR TYPE OF ARRAY ( = 1 FOR 64K CHIPS
            ;4631 ;         = 0 FOR 256K CHIPS)
            ;4632 ;
            ;4633 ; 9 LOCATION UNDER TEST
            ;4634 ;
            ;4635 ; A RECEIVED LOWER LONGWORD
            ;4636 ;
            ;4637 ; B RECEIVED UPPER LONGWORD
            ;4638 ;
            ;4639 ; C EXPECTED UPPER/LOWER DATA
            ;4640 ;
            ;4641 ; D CNFG REGISTER C/D ADDRESS FOR CONTROLLER UNDER TEST
            ;4642 ;
            ;4643 ;         E       I/O BASE ADDRESS OF MS780-E/H CURRENTLY INDER TEST
            ;4644 ;
            ;4645 ;--
            ;4646 ;*************************************************************
            ;4647 =0
U 118E, 0000.003D,0180,0800,0000,10A3  ;4648 TEST.6: NEWTST     ; Increment test number
U 118F, 0018.0038,0D80,09A0,0000,10A0  ;4649  RC[04]_K[.3]     ; Clear location for type of memory flag
U 10A0, 0000.003D,0180,0800,0000,10D8  ;4650 =00 CALL[FIND.MS780E]   ; Any MS780-E/Hs on the SBI?
U 10A1, 0000.003C,0180,0800,0000,12E6  ;4651  J/TEST.6.EXIT    ; No MS780-E/Hs found
U 10A2, 0000.003C,0180,0800,0000,10A4  ;4652  NOP
            ;4653 =
            ;4654 =00
            ;4655 RESTART.TEST.6:     ; Entry Point for second controller
U 10A4, 0000.003D,0180,0800,0000,10C6  ;4656  CALL[START.TEST]  ; ...or next MS780-E/H
            ;4657       ; ...Configure Controllers
U 10A5, 0000.003C,0180,0800,0000,12E6  ;4658  J/TEST.6.EXIT    ; No more controllers or
            ;4659       ; ...MS780-E/Hs found
U 10A6, 0000.003C,0180,0800,0000,10A8  ;4660  NOP
            ;4661 =
U 10A8, 0000.003D,0180,0800,0000,11E4  ;4662 =00 CALL[64K.OR.256K]  ; Find the type of chips that are on the
            ;4663       ; ...arrays: 64K or 256K chips
```

```
            ;4664  .PAGE
            ;4665 ;
            ;4666 ;////////////////////////////////////////////////////////////////////////////////
            ;4667 ;
U 10A9, 0000,003D,8580,0800,0084,70C0  ;4668  ERROR1[.C]   ; Mixed boards on this Controller
            ;4669       ; ...both 64K and 256K chips !!
            ;4670       ; ...or Missing Arrays !!
            ;4671 ;
            ;4672 ;////////////////////////////////////////////////////////////////////////////////
            ;4673 ;
            ;4674 ; ERROR LOGOUT:
            ;4675 ;
            ;4676 ; DATA: I/O BASE ADDRESS OF MS780-E/H UNDER TEST
            ;4677 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4678 ;  NOT USED
            ;4679 ;  NOT USED
            ;4680 ;  NOT USED
            ;4681 ;  NOT USED
            ;4682 ;  NOT USED
            ;4683 ;  NOT USED
            ;4684 ;  NOT USED
            ;4685 ;  NOT USED
            ;4686 ;  ERROR CODE = 3 (I.E., MIXED OR MISSING ARRAYS)
            ;4687 ;
            ;4688 ;
            ;4689 ;////////////////////////////////////////////////////////////////////////////////
            ;4690 ;
            ;4691  J/T.6.LOAD.REGS,  ;
U 10AA, 0018,0038,0580,09A0,0000,12D6  ;4692   RC[04]_K[.1]  ; Found 64K chips on the array
U 10AB, 0003,003C,0180,09A0,0000,12D6  ;4693  RC[04]_0   ; Found 256K chips on the array
            ;4694 T.6.LOAD.REGS:
U 12D6, 0018,0038,3180,0800,0200,1190  ;4695  VA_K[.40]   ; Point to test data in cache
U 1190, 0000,003D,0180,0800,0000,125F  ;4696 =0 CALL[LOAD.TEMP.REGS]  ; Load test data in temporary registers
            ;4697       ; ...T1 through T7
U 1191, 0000,003C,0180,0800,0000,1192  ;4698  NOP
            ;4699
            ;4700 ;
            ;4701 ;////////////////////////////////////////////////////////////////////////////////
            ;4702 ;**
            ;4703 ;
            ;4704 ; SUBTEST 1
            ;4705 ;
            ;4706 ;--
            ;4707 ;////////////////////////////////////////////////////////////////////////////////
            ;4708
U 1192, 0000,003D,0180,0800,0000,11D8  ;4709 =0 SUBTEST    ; Increment subtest number
U 1193, 0003,003C,0180,09C8,0000,12D7  ;4710  RC[9]_0    ; Quad-Word Location 0 will be
            ;4711       ; ...under test
U 12D7, 0838,0038,B980,0800,0000,12D8  ;4712  D_ALU.LEFT,ALU_K[.19]  ; Load x32 in R[1].  Pointer to data in
U 12D8, 0001,003C,0180,0A88,0000,12D9  ;4713  R[1]_D    ; ...Temporary registers
U 12D9, 0018,0038,D580,0A80,0000,1194  ;4714  R[0]_K[.6]   ; Load x06 in Register R[0].  This will
            ;4715       ; ...be the Temporary Register Counter
            ;4716 =0
            ;4717 TEST.6.LOOP1:
U 1194, 0000,003D,0180,0800,0000,11D0  ;4718  ERLOOP    ; Set up "Loop on Error" mechanism
```

```
U 1195. 0000.003C.0180.0800.0000.1196  ;4719  NOP
U 1196. 0000.003D.0180.0800.0000.10D0  ;4720 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1197. 0000.003C.0180.0800.0000.1198  ;4721  NOP
U 1198. 0818.0039.0D80.0800.0000.11F0  ;4722 =0 SET.DIAG.MODE[.3]  ; Set ECC bypass Diagnostic Mode
U 1199. 0000.003C.0180.0A08.0082.12DA  ;4723  SC_R[1]    ; Point the SC to the data in Temp Regs
U 12DA. 0010.0038.0180.0948.0200.12DB  ;4724  VA_RC[09]   ; Point the VA to location under test
U 12DB. 0000.003C.01F0.2400.0000.12DC  ;4725  Q_ID(SC)   ; Get the test Pattern
U 12DC. 0801.203C.0180.09E0.0000.12DD  ;4726  RC[0C]_Q,D_ALU   ; Get the test pattern in the D Reg and
        ;4727       ; ...save it for error logout
U 12DD. 0000.003C.0180.A000.0000.10AC  ;4728  MCT/EXTWRITE.P   ; Extended write to Quad-Word location
        ;4729       ; ...0
U 10AC. 0000.003D.0180.0800.0000.1205  ;4730 =00 CHECK.INTERRUPT   ; Check for occurrence of Unexpected
        ;4731       ; ...Interrupts
U 10AD. 0000.003C.0180.0800.0000.119A  ;4732  J/TEST.6.NOINTR1  ; No interrupt on Write
```

```
            ;4733  .PAGE
            ;4734 ;
            ;4735 ;////////////////////////////////////////////////////////////////////////////////
            ;4736 ;
U 10AE, 0000,003D,8580,0800,0084,70C2  ;4737 =0 ERROR2[.C]   ; Unexpected Interrupt Occurred
            ;4738       ; ...on Write to the array
            ;4739 ;
            ;4740 ;////////////////////////////////////////////////////////////////////////////////
            ;4741 ;
            ;4742 ; ERROR LOGOUT:
            ;4743 ;
            ;4744 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4745 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4746 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4747 ;  NOT USED
            ;4748 ;  NOT USED
            ;4749 ;  QUAD-WORD LOCATION UNDER TEST
            ;4750 ;  ID VECTOR REGISTER
            ;4751 ;  ID SBIER REGISTER
            ;4752 ;  ID FAULT REGISTER
            ;4753 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;4754 ;        0 = NO INTERRUPT OCCURRED )
            ;4755 ;  TYPE OF ARRAY FLAG ( 1 = 64K
            ;4756 ;          0 = 256K )
            ;4757 ;  NOT USED
            ;4758 ;
            ;4759 ;
            ;4760 ;////////////////////////////////////////////////////////////////////////////////
            ;4761 ;
U 10AF, 0000,003C,0180,0800,0000,119A  ;4762  NOP
            ;4763 =
            ;4764 =0
            ;4765 TEST.6.NOINTR1:
U 119A, 0000,003D,0180,0800,0000,11D0  ;4766  ERLOOP    ; Set the error loop mechanism
U 119B, 0000,003C,0180,0800,0000,119C  ;4767  NOP
U 119C, 0000,003D,0180,0800,0000,10D0  ;4768 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 119D, 0000,003C,0180,0800,0000,119E  ;4769  NOP
U 119E, 0818,0039,0D80,0800,0000,11F0  ;4770 =0 SET.DIAG.MODE[.3]  ; Set ECC Bypass Diagnostic Mode
U 119F, 0010,0038,0180,0948,0200,11A0  ;4771  VA_RC[09]   ; Point to Lower Longword
U 11A0, 0000,003D,8980,0800,0084,721C  ;4772 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 11A1, 0810,0038,0180,0960,0000,12DE  ;4773  D_RC[0C]   ; Get data to do another write to
            ;4774       ; ...the location
U 12DE, 0000,003C,0180,A000,0000,11A2  ;4775  MCT/EXTWRITE.P   ; Do the Extended write
U 11A2, 0000,003D,8580,0800,0084,721C  ;4776 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 11A3, 0000,003C,0180,C800,0000,10B0  ;4777  D[LONG]_CACHE.P   ; Read Lower Longword
            ;4778 =00 RC[0A]_D,    ; Save the received data
U 10B0, 0001,003D,0180,09D0,0000,1212  ;4779   CHECK.UTRAP  ; Was there a uTrap (Tm0 or RDS)?
U 10B1, 0000,003C,0180,0800,0000,10B4  ;4780  J/TEST.6.NOTRP1   ; OK.  No uTrap Occurred
```

```
            ;4781   .PAGE
            ;4782 ;
            ;4783 ;////////////////////////////////////////////////////////////////////////////////
            ;4784 ;
U 10B2, 0000,003D,8580,0800,0084,70C2  ;4785 =0 ERROR2[.C]    ; uTrap on Read from Lower Longword
            ;4786       ; ...of Qwad-Word location 0
            ;4787 ;
            ;4788 ;////////////////////////////////////////////////////////////////////////////////
            ;4789 ;
            ;4790 ; ERROR LOGOUT:
            ;4791 ;
            ;4792 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4793 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4794 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4795 ;  NOT USED
            ;4796 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4797 ;  QUAD-WORD LOCATION UNDER TEST
            ;4798 ;  NOT USED
            ;4799 ;  NOT USED
            ;4800 ;  NOT USED
            ;4801 ;  NOT USED
            ;4802 ;  TYPE OF ARRAY FLAG ( 1 = 64K CHIPS
            ;4803 ;           0 = 256K CHIPS )
            ;4804 ;  NOT USED
            ;4805 ;
            ;4806 ;
            ;4807 ;////////////////////////////////////////////////////////////////////////////////
            ;4808 ;
U 10B3, 0000,003C,0180,0800,0000,10B4  ;4809  NOP
            ;4810 =00
            ;4811 TEST.6.NOTRP1:
U 10B4, 0000,003D,0180,0800,0000,1205  ;4812  CHECK.INTERRUPT    ; Any unexpected interrupts occurred?
U 10B5, 0000,003C,0180,0800,0000,10B7  ;4813  J/TEST.6.NOINTR2   ; OK. No Unexpected Interrupt
```

```
            ;4814  .PAGE
            ;4815 ;
            ;4816 ;////////////////////////////////////////////////////////////////////////////////
            ;4817 ;
U 10B6, 0000,003D,8580,0800,0084,70C2  ;4818 =0 ERROR2[.C]   ; Unexpected Interrupt Occurred
            ;4819      ; ...on Write to the array
            ;4820 ;
            ;4821 ;////////////////////////////////////////////////////////////////////////////////
            ;4822 ;
            ;4823 ; ERROR LOGOUT:
            ;4824 ;
            ;4825 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4826 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4827 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4828 ;  NOT USED
            ;4829 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4830 ;  QUAD-WORD LOCATION UNDER TEST
            ;4831 ;  ID VECTOR REGISTER
            ;4832 ;  ID SBIER REGISTER
            ;4833 ;  ID FAULT REGISTER
            ;4834 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;4835 ;       0 = NO INTERRUPT OCCURRED )
            ;4836 ;  TYPE OF ARRAY FLAG ( 1 = 64K
            ;4837 ;         0 = 256K )
            ;4838 ;  NOT USED
            ;4839 ;
            ;4840 ;
            ;4841 ;////////////////////////////////////////////////////////////////////////////////
            ;4842 ;
            ;4843 TEST.6.NOINTR2:
U 10B7, 0810,0038,0180,0960,0000,12DF  ;4844  D_RC[0C]    ; Get the expected data pattern
            ;4845  ALU_D.XOR.RC[0A],  ; Is the expected equal to the
U 12DF, 0011,0020,0180,0950,0010,12E0  ;4846   CLK.UBCC   ; ...received data?
U 12E0, 0000,013C,0180,0800,0000,11A4  ;4847  Z?     ; ...
```

```
            ;4848  .PAGE
            ;4849 ;
            ;4850 ;////////////////////////////////////////////////////////////////////////////////
            ;4851 ;
U 11A4. 0000.003D.8580.0800.0084.70C2  ;4852 =0 ERROR2[.C]   ; Wrong data returned from Lower
            ;4853       ; ...Longword of Quad-Word location 0
            ;4854 ;
            ;4855 ;////////////////////////////////////////////////////////////////////////////////
            ;4856 ;
            ;4857 ; ERROR LOGOUT:
            ;4858 ;
            ;4859 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4860 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4861 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4862 ;  NOT USED
            ;4863 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4864 ;  QUAD-WORD LOCATION UNDER TEST
            ;4865 ;  NOT USED
            ;4866 ;  NOT USED
            ;4867 ;  NOT USED
            ;4868 ;  NOT USED
            ;4869 ;  TYPE OF ARRAY FLAG ( 1 = 64K CHIPS
            ;4870 ;                0 = 256K CHIPS )
            ;4871 ;  NOT USED
            ;4872 ;
            ;4873 ;
            ;4874 ;////////////////////////////////////////////////////////////////////////////////
            ;4875 ;
U 11A5. 0000.003C.0180.0800.0000.11A6  ;4876  NOP
U 11A6. 0000.003D.0180.0800.0000.11D0  ;4877 =0 ERLOOP     ; Set the error loop mechanism
U 11A7. 0000.003C.0180.0800.0000.11A8  ;4878  NOP
U 11A8. 0000.003D.0180.0800.0000.10D0  ;4879 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 11A9. 0000.003C.0180.0800.0000.11AA  ;4880  NOP
U 11AA. 0818.0039.0D80.0800.0000.11F0  ;4881 =0 SET.DIAG.MODE[.3]  ; Set ECC Bypass Diagnostic Mode
U 11AB. 0018.0038.1180.0800.0200.11AC  ;4882  VA_K[.4]    ; Point to Upper Longword
U 11AC. 0000.003D.8980.0800.0084.721C  ;4883 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 11AD. 0810.0038.0180.0960.0000.12E1  ;4884  D_RC[0C]    ; Get data to do another write to
            ;4885       ; ...the location
U 12E1. 0000.003C.0180.A000.0000.11AE  ;4886  MCT/EXTWRITE.P    ; Do the Extended write
U 11AE. 0000.003D.8580.0800.0084.721C  ;4887 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 11AF. 0000.003C.0180.C800.0000.10B8  ;4888  D[LONG]_CACHE.P    ; Read Upper Longword
            ;4889 =00 RC[0B]_D,    ; Save the received data
U 10B8. 0001.003D.0180.09D8.0000.1212  ;4890   CHECK.UTRAP  ; Was there a uTrap (Tm0 or RDS)?
U 10B9. 0000.003C.0180.0800.0000.10BC  ;4891  J/TEST.6.NOTRP2    ; OK.  No uTrap Occurred
```

```
            ;4892  .PAGE
            ;4893 ;
            ;4894 ;////////////////////////////////////////////////////////////////////////////////
            ;4895 ;
U 10BA, 0000,003D,8580,0800,0084,70C2  ;4896 =0 ERROR2[.C]    ; uTrap on Read from Lower Longword
            ;4897       ; ...of Qwad-Word location 0
            ;4898 ;
            ;4899 ;////////////////////////////////////////////////////////////////////////////////
            ;4900 ;
            ;4901 ; ERROR LOGOUT:
            ;4902 ;
            ;4903 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4904 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4905 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4906 ;  RECEIVED DATA FROM UPPER LONGWORD
            ;4907 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4908 ;  QUAD-WORD LOCATION UNDER TEST
            ;4909 ;  NOT USED
            ;4910 ;  NOT USED
            ;4911 ;  NOT USED
            ;4912 ;  NOT USED
            ;4913 ;  TYPE OF ARRAY FLAG ( 1 = 64K CHIPS
            ;4914 ;          0 = 256K CHIPS )
            ;4915 ;  NOT USED
            ;4916 ;
            ;4917 ;
            ;4918 ;////////////////////////////////////////////////////////////////////////////////
            ;4919 ;
U 10BB, 0000,003C,0180,0800,0000,10BC  ;4920  NOP
            ;4921 =
            ;4922 =00
            ;4923 TEST.6.NOTRP2:
U 10BC, 0000,003D,0180,0800,0000,1205  ;4924  CHECK.INTERRUPT   ; Any unexpected interrupts occurred?
U 10BD, 0000,003C,0180,0800,0000,10BF  ;4925  J/TEST.6.NOINTR3  ; OK. No Interrupt Occurred
```

```
            ;4926  .PAGE
            ;4927 ;
            ;4928 ;////////////////////////////////////////////////////////////////////////////////
            ;4929 ;
U 10BE, 0000,003D,8580,0800,0084,70C2  ;4930 =0 ERROR2[.C]    ; Unexpected Interrupt Occurred
            ;4931       ; ...on Read from the Upper Longword
            ;4932 ;
            ;4933 ;////////////////////////////////////////////////////////////////////////////////
            ;4934 ;
            ;4935 ; ERROR LOGOUT:
            ;4936 ;
            ;4937 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4938 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4939 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4940 ;  RECEIVED DATA FROM UPPER LONGWORD
            ;4941 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4942 ;  QUAD-WORD LOCATION UNDER TEST
            ;4943 ;  ID VECTOR REGISTER
            ;4944 ;  ID SBIER REGISTER
            ;4945 ;  ID FAULT REGISTER
            ;4946 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;4947 ;        0 = NO INTERRUPT OCCURRED )
            ;4948 ;  TYPE OF ARRAY FLAG ( 1 = 64K
            ;4949 ;          0 = 256K )
            ;4950 ;  NOT USED
            ;4951 ;
            ;4952 ;
            ;4953 ;////////////////////////////////////////////////////////////////////////////////
            ;4954 ;
            ;4955 TEST.6.NOINTR3:
U 10BF, 0810,0038,0180,0960,0000,12E2  ;4956  D_RC[0C]    ; Get the expected data pattern
            ;4957  ALU_D.XOR.RC[0B],   ; Is the expected equal to the
U 12E2, 0011,0020,0180,0958,0010,12E3  ;4958   CLK.UBCC   ; ...received data?
U 12E3, 0000,013C,0180,0800,0000,11B0  ;4959  Z?     : ...
```

```
            ;4960  .PAGE
            ;4961 ;
            ;4962 ;////////////////////////////////////////////////////////////////////////////////
            ;4963 ;
U 11B0, 0000,003D,8580,0800,0084,70C2  ;4964 =0 ERROR2[.C]   ; Wrong data returned from Upper
            ;4965      ; ...Longword of Quad-Word location 0
            ;4966 ;
            ;4967 ;////////////////////////////////////////////////////////////////////////////////
            ;4968 ;
            ;4969 ; ERROR LOGOUT:
            ;4970 ;
            ;4971 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4972 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4973 ;  EXPECTED DATA FROM LOWER/UPPER LONGWORD
            ;4974 ;  RECEIVED DATA FROM UPPER LONGWORD
            ;4975 ;  RECEIVED DATA FROM LOWER LONGWORD
            ;4976 ;  QUAD-WORD LOCATION UNDER TEST
            ;4977 ;  NOT USED
            ;4978 ;  NOT USED
            ;4979 ;  NOT USED
            ;4980 ;  NOT USED
            ;4981 ;  TYPE OF ARRAY FLAG ( 1 = 64K CHIPS
            ;4982 ;          0 = 256K CHIPS )
            ;4983 ;  NOT USED
            ;4984 ;
            ;4985 ;
            ;4986 ;////////////////////////////////////////////////////////////////////////////////
            ;4987 ;
U 11B1, 0800,003C,0180,0A00,0000,12E4  ;4988  D_R[0]    ; Get the register counter
            ;4989  R[0]_D-K[.1],   ; Decrement the register
U 12E4, 0019,0000,0580,0A80,0010,12E5  ;4990   CLK.UBCC  ; See if it is zero.
            ;4991  Z?,     ; ...
U 12E5, 0800,013C,0180,0A08,0000,11B2  ;4992   D_R[1]    ; Get the Temporary Register Pointer
            ;4993 =0 J/TEST.6.LOOP1,   ; Not finished yet
U 11B2, 0019,0014,0580,0A88,0000,1194  ;4994   R[1]_D+K[.1]  ; Increment the Temporary Register
            ;4995      ; ...Pointer
U 11B3, 0000,003C,0180,0800,0000,11B4  ;4996  NOP     ; Finished SUBTEST 1
            ;4997 ;
            ;4998 ;////////////////////////////////////////////////////////////////////////////////
            ;4999 ;**
            ;5000 ;
            ;5001 ; SUBTEST 2
            ;5002 ;
            ;5003 ;--
            ;5004 ;////////////////////////////////////////////////////////////////////////////////
            ;5005 ;
U 11B4, 0000,003D,0180,0800,0000,11D8  ;5006 =0 SUBTEST    ; Increment subtest number
U 11B5, 0F00,003C,0180,0800,0000,11B6  ;5007  D_0     ; Point to location 0
U 11B6, 0000,003D,0180,0800,0000,1267  ;5008 =0 CALL[DATA.CHECK]  ; Test Longword location 0
U 11B7, 0818,0038,1180,0800,0000,11B8  ;5009  D_K[.4]    ; Point to locaiton 4
U 11B8, 0000,003D,0180,0800,0000,1267  ;5010 =0 CALL[DATA.CHECK]  ; Test Longword location 4
U 11B9, 0000,003C,0180,0800,0000,10A4  ;5011  J/RESTART.TEST.6  ; Check for more Controllers or
            ;5012      ; ...MS780-E/Hs
            ;5013 TEST.6.EXIT:
U 12E6, 0000,003C,0180,0800,0000,11BA  ;5014  NOP     ; Finished TEST 6
```

```
          ;5015
```

```
            ;5016  .PAGE "TEST 18B RESERVED"
            ;5017 =0
U 11BA, 0000,003D,0180,0800,0000,10A3  ;5018 T.18B: NEWTST
U 11BB, 0000,003C,0180,0800,0000,11BC  ;5019  NOP
            ;5020
```

```
            ;5021   .PAGE "TEST 18C RESERVED"
            ;5022 =0
U 11BC. 0000.003D.0180.0800.0000.10A3  ;5023 T.18C: NEWTST
U 11BD. 0000.003C.0180.0800.0000.12E7  ;5024  NOP
            ;5025
            ;5026
```

```
            ;5027  .PAGE "DUMMY NEWTST"
U 12E7, 0000,003D,0180,0800,0000,10A3  ;5028 DUM: NEWTST
```

```
:                          Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2809= 2810= 2811= 2814=
U 1008  2410= 2412= 2414= 2419= 2260= 2261= 2262= 2263=
U 1010  2433= 2435= 2436= 2440= 2456= 2457= 2458= 2466=
U 1018  2468= 2470= 2472= 2473= 2399= 2400= 2401= 2402=
U 1020  2516= 2517= 2518= 2520= 2564= 2565= 2566= 2568=
U 1028  2966= 2967= 2968= 2969= 2970= 2971= 2972= 2973=
U 1030  3260= 3261= 3265= 3284= 3300= 3301= 3306= 3321=
U 1038  3324= 3325= 3330= 3349= 3425= 3426= 3433= 3435=
U 1040  3475= 3476= 3477= 3478= 3480= 3481= 3482= 1902
U 1048  3486= 3491= 3503= 3504= 1912= 1913= 2021: 1903
U 1050  3633= 3638= 3647= 1904 3664= 3665= 3669= 3686=
U 1058  1965= 1967= 2067: 1906 1978= 1979= 2101: 1907
U 1060  3689= 3690= 3695= 3713= 3724= 3725= 3730= 3748=
U 1068  3771= 3772= 3773= 1914 3860= 3861= 3862= 1915
U 1070  3879= 3880= 3882= 1916 4042= 4044= 4045= 1917
U 1078  4050= 4051= 4052= 1918 4092= 4093= 4094= 4095=
U 1080  4175= 4176= 4177= 4178= 4306= 4307= 4308= 1919
U 1088  4313= 4314= 4315= 1920 4340= 4341= 4346= 4368=
U 1090  4380= 4382= 4387= 4406= 4409= 4410= 4415= 4438=
U 1098  4474= 4476= 4481= 4500= 4503= 4504= 4509= 4532=
U 10A0  4650= 4651= 4652= 1954 4656= 4658= 4660= 1955
U 10A8  4662= 4668= 4692= 4693= 4730= 4732= 4737= 4762=
U 10B0  4779= 4780= 4785= 4809= 4812= 4813= 4818= 4844=
U 10B8  4890= 4891= 4896= 4920= 4924= 4925= 4930= 4956=
U 10C0  2039= 2040= 2085= 2086= 2208= 2209= 2388= 2389=
U 10C8  2391= 2396= 2626= 2631= 2636= 2641= 2646= 2647=
U 10D0  2691= 2692= 2769= 2770= 2848= 2849= 2879= 2880=
U 10D8  2945= 2946= 2954= 2955= 2956= 2958= 2961= 2962=
U 10E0  3003= 3004= 3005= 3006= 3013= 3014= 3020= 3022=
U 10E8  3025= 3026= 3045= 3046= 3048= 3049= 3138= 3139=
U 10F0  3142= 3143= 3154= 3155= 3158= 3159= 3186= 3187=
U 10F8  3198= 3199= 3200= 3201= 3251= 3252= 3253= 3254=
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895: 1956 3287= 3288= 1896: 1897: 1898: 1899:
U 1110  3291= 3292= 3293= 3294= 3295= 3296= 3359= 3375=
U 1118  3379= 3380= 3473= 3474= 3508= 3510= 3516= 3517=
U 1120  3524= 3525= 3530= 3532= 3631= 3632= 3651= 3652=
U 1128  3658= 3659= 3660= 3661= 3662= 3663= 3716= 3717=
U 1130  3718= 3719= 3722= 3723= 3752= 3753= 3760= 3761=
U 1138  3762= 3763= 3764= 3765= 3766= 3767= 3769= 3770=
U 1140  3858= 3859= 3864= 3868= 3871= 3873= 3876= 3878=
U 1148  4040= 4041= 4059= 4060= 4064= 4065= 4067= 4068=
U 1150  4071= 4072= 4081= 4082= 4083= 4084= 4085= 4086=
U 1158  4087= 4088= 4090= 4091= 4102= 4124= 4129= 4130=
U 1160  4137= 4138= 4153= 4154= 4164= 4165= 4166= 4167=
U 1168  4168= 4169= 4170= 4171= 4173= 4174= 4185= 4205=
U 1170  4212= 4213= 4304= 4305= 4318= 4320= 4327= 4328=
U 1178  4329= 4330= 4331= 4332= 4371= 4372= 4375= 4376=
U 1180  4377= 4378= 4447= 4464= 4465= 4466= 4469= 4470=
U 1188  4471= 4472= 4541= 4558= 4564= 4565= 4648= 4649=
U 1190  4696= 4698= 4709= 4710= 4718= 4719= 4720= 4721=
U 1198  4722= 4723= 4766= 4767= 4768= 4769= 4770= 4771=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   4772= 4773= 4776= 4777= 4852= 4876= 4877= 4878=
U 11A8   4879= 4880= 4881= 4882= 4883= 4884= 4887= 4888=
U 11B0   4964= 4988= 4994= 4996= 5006= 5007= 5008= 5009=
U 11B8   5010= 5011= 5018= 5019= 5023= 5024= 1957 1958
U 11C0   1959 1960 1961 1962 1963 1964 1968 1969
U 11C8   1970 1971 1972 1973 1974 1975 1976 1977
U 11D0   1996 2019 2020 2065 2066 2149 2150 2151
U 11D8   2169 2171 2197 2198 2199 2200 2201 2202
U 11E0   2203 2204 2206 2207 2254 2255 2256 2258
U 11E8   2259 2397 2398 2423 2443 2444 2445 2452
U 11F0   2592 2593 2594 2595 2596 2597 2621 2669
U 11F8   2693 2694 2695 2696 2714 2715 2716 2717
U 1200   2735 2736 2737 2738 2764 2798 2800 2801
U 1208   2802 2804 2805 2806 2807 2808 2815 2816
U 1210   2817 2818 2841 2842 2843 2844 2846 2847
U 1218   2874 2875 2877 2878 2906 2907 2908 2947
U 1220   2948 2949 2950 2959 2960 2964 2965 2978
U 1228   2979 2981 2982 2983 2985 2990 2991 2992
U 1230   2994 2999 3001 3002 3010 3011 3015 3016
U 1238   3017 3018 3019 3027 3028 3029 3031 3032
U 1240   3033 3034 3035 3036 3041 3043 3044 3047
U 1248   3072 3073 3074 3075 3103 3104 3106 3107
U 1250   3108 3134 3136 3137 3141 3145 3146 3147
U 1258   3148 3150 3151 3152 3153 3156 3157 3185
U 1260   3190 3191 3193 3194 3195 3196 3197 3242
U 1268   3243 3244 3245 3246 3247 3248 3255 3256
U 1270   3257 3258 3259 3290 3297 3298 3351 3353
U 1278   3354 3376 3377 3378 3419 3420 3421 3422
U 1280   3423 3437 3439 3505 3506 3507 3511 3512
U 1288   3513 3514 3515 3518 3519 3520 3521 3522
U 1290   3523 3526 3527 3528 3529 3653 3654 3720
U 1298   3721 3749 3750 3751 3768 3869 3874 4062
U 12A0   4073 4074 4075 4076 4077 4078 4079 4125
U 12A8   4126 4127 2127: 4128 4139 4141 4143 4144
U 12B0   4145 4146 4147 4148 4149 4150 4156 4157
U 12B8   4158 4159 4160 4161 4162 4206 4207 4208
U 12C0   4209 4210 4211 4216 4321 4322 4324 4333
U 12C8   4335 4336 4338 4374 4441 4442 4468 4535
U 12D0   4536 4559 4560 4562 4563 4568 4695 4712
U 12D8   4713 4714 4724 4725 4726 4728 4775 4846
U 12E0   4847 4886 4958 4959 4990 4992 5014 5028
U 12E8 - 13EF Unused
U 13F0   2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8   2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR3F	760
Used	760
Remaining	

Total microwords used in memory U: 760
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR3F.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 0231
ESKAR40.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1848 Micro Trap Handler and LSI-11 Support Routines
; 1849   Micro Trap Handler
; 1928 Micro Monitor Interface Routines
; 1929   Newtest Routine
; 1985   Error Loop Routine
; 2002   Error 1 Routine
; 2027   ERROR1 WITH PARAMETER
; 2050   Error 2 Routine
; 2076   ERROR 2 WITH PARAMETER
; 2097   Micro-Monitor Call
; 2111   Reserved Control Store
; 2138   Set Argument Count
; 2161   Increment Subtest Number
; 2181 Diagnostic Specific Subroutines
; 2182   Select Controller
; 2228   START TEST
; 2437   SELECT LOWER CONTROLLER
; 2485   SELECT UPPER CONTROLLER
; 2532   SBI Unjam Routine
; 2581   RESET SUBTEST NUMBER
; 2603   Initialize the SBI
; 2630   Disable Cache Routine
; 2651   Enable Cache
; 2672   Wait Loop Routine
; 2705   Check for Interrupt Routine
; 2752   CHECK UTRAP
; 2784   CHECK SBI.ERR
; 2814   Set Trap Mask
; 2842   FIND MS780-E MEMORY
; 2984   Generate IO Address
; 3009   Set Starting Address
; 3044   FIND CPU ID
; 3100   CHECK INTERRUPT VECTOR ROUTINE
; 3153   ENABLE NEXT MS780-E
; 3205   INIT SUBROUTINE
; 3236   SET PSL SUBROUTINE
; 3249   CLEAR TIME OUT BIT SUBROUTINE
; 3262   CLEAR FAULT REGISTER
; 3275   CLEAR ECC IN SBI.ERR
; 3288   ENABLE ECC, CRD/RDS INTRPTS
; 3315 TEST 18D SBI FORCED FAULT TESTS
; 5163 TEST 18E CPU AND MEMORY MULT XMIT FLT
; 5547 TEST 18F RESERVED
; 5552 TEST 190 RESERVED
; 5557 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        :1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR40
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; MODIFIED BY: PAUL HAKALA
;1836 ;
;1837 ; 01-00   23-NOV-82
;1838 ;  Modified test ESKAR40 Test 18E to go look for another
;1839 ;  MS780-E on the SBI (Cpu and Mem Mult Trans Fit Tst)
;1840 ;
;1841 ;
;1842 ;
;1843 ;
;1844 ;*****************************************************************************
;1845
```

```
;1846  .PAGE
;1847
;1848  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1849  .TOC "    Micro Trap Handler"
;1850 ;**
;1851 ; FUNCTIONAL DESCRIPTION:
;1852 ;
;1853 ; This routine is responsible for 'catching' micro-traps
;1854 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1855 ; contains the Trap-Expected Mask. If the specified bit is set in
;1856 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1857 ; an error call is made.
;1858 ;
;1859 ; INPUT PARAMETERS:
;1860 ;
;1861 ; R[0F] - Expected Trap Mask
;1862 ;  Trap Code Function
;1863 ;  --------      --------
;1864 ;       0  System Init
;1865 ;       1  Data Unaligned
;1866 ;       2  Cross Page
;1867 ;       3  TB Modify
;1868 ;       4  TB Protection
;1869 ;       6  TB Miss
;1870 ;       7  TB Parity
;1871 ;       8  Cache Parity
;1872 ;       9  Reserved Floating Operand
;1873 ;       C  Read Data Substitute
;1874 ;       D  Time out
;1875 ;       F  Control Store Parity Error
;1876 ;      10  Odd Address
;1877 ;
;1878 ; OUTPUT PARAMETERS:
;1879 ;
;1880 ; R[0F] - Mask bit is cleared.
;1881 ;
;1882 ; DATA: Micro PC of Micro Trap
;1883 ;  Trap Code (See Above)
;1884 ;  Fault Status Register
;1885 ;  SBI Error Register
;1886 ;  Timeout Address Register
;1887 ;  Maintenance Register
;1888 ;  Cache Parity Register
;1889 ;  TB Error Register 1
;1890 ;  TB Error Register 0
;1891 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1892 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1893 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1894 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1895 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1896 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1897 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1898 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1899 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1900 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000.003C.8580.0800.0084.7001  ;1901 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000.003C.8980.0800.0084.7001  ;1902 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000.003C.6580.0800.0084.7001  ;1903 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000.003C.6180.0800.0084.7003  ;1904 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1905
U 1001, 0000.003C.81F0.2C00.0000.1033  ;1906 trph0: q_id[ustack]  ; Get upc of trap
U 1033, 0C00.003C.01E0.0800.0000.1037  ;1907  q_d.d_q   ; Setup to put it back on stack
U 1037, 0C00.003C.81E0.3C00.0000.103B  ;1908  id[ustack]_d,q_d.d_q ; Put upc back onto micro stack
U 103B, 0000.003C.01C0.0A78.0000.103F  ;1909  q_r[0f]   ; Get the Expected Trap Mask
        ;1910  alu_q.andnot.mask,
U 103F, 0001.2024.0180.0800.0010.1043  ;1911   clk.ubcc ; Was trap expected?
U 1043, 0000.013C.0180.0800.0000.1002  ;1912  z?
        ;1913 =0 r[0f]_alu.  ; It was, clear the mask and return
        ;1914       alu_q.and.mask, ; ...
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1915       return0  ; ...
U 1003, 0018.0038.1D80.09E8.0000.104C  ;1916 trph1: rc[0d]_sc  ; Output the Trap Code
U 104C, 0000.003D.D980.0800.0084.71ED  ;1917 =0 setrcf[.9]  ; Set output count
U 104D, 0000.003C.49F0.2C00.0000.1047  ;1918  q_id[tber0]  ; Output all the error registers
U 1047, 0001.203C.4DF0.2DB0.0000.104B  ;1919  rc[6]_q,q_id[tber1] ; ...
U 104B, 0001.203C.65F0.2DB8.0000.104F  ;1920  rc[7]_q,q_id[sbi.err] ; ...
U 104F, 0001.203C.6DF0.2DD8.0000.1053  ;1921  rc[0b]_q,q_id[fault] ; ...
U 1053, 0001.203C.75F0.2DE0.0000.1057  ;1922  rc[0c]_q,q_id[maint] ; ...
U 1057, 0001.203C.79F0.2DC8.0000.105B  ;1923  rc[9]_q,q_id[parity] ; ...
U 105B, 0001.203C.69F0.2DC0.0000.105F  ;1924  rc[8]_q,q_id[time.addr]; ...
U 105F, 0001.203C.81F0.2DD0.0000.1000  ;1925  rc[0a]_q,q_id[ustack] ; ...
U 1000, 0001.203D.0180.09F0.0000.11E5  ;1926 1000: ERROR1,rc[0e]_q  ; Report the error
```

```
                ;1927  .PAGE
                ;1928  .TOC " Micro Monitor Interface Routines"
                ;1929  .TOC "   Newtest Routine"
                ;1930 ;++
                ;1931 ; FUNCTIONAL DESCRIPTION:
                ;1932 ;
                ;1933 ; This routine Initializes the following registers:
                ;1934 ;  PSL    to 1F
                ;1935 ;  CES    to 0
                ;1936 ;  ACC.CS    to disable accellerator
                ;1937 ;  SIR    to 0
                ;1938 ;  CLK.CS    to stop interval timer
                ;1939 ;  TBER0    to disable the TB
                ;1940 ;  SBI.MAINT to     0
                ;1941 ;  SBI.ERR    to 0
                ;1942 ;  FAULT    to 0
                ;1943 ;  COMP    to 0
                ;1944 ;  RC[0E]    to  0
                ;1945 ;  R[0F]    to 0
                ;1946 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1947 ; A SCOPE call is then made to the Monitor.
                ;1948 ;
                ;1949 ; CALLING CONSTRAINT:
                ;1950 ;
                ;1951 ; =0
                ;1952 ;
                ;1953 ; SIDE EFFECTS:
                ;1954 ;
                ;1955 ; D Reg is destroyed
                ;1956 ; SC is destroyed
                ;1957 ;--
U 1063. 0000.003C.65F8.0800.0084.7067   ;1958 SCOPE: sc_k[.10],q_0   ; Set IPL to 1F
U 1067. 0818.0038.8D80.0800.0000.106B   ;1959  d_k[.1f]   ; D=1F
U 106B. 0D00.003C.0180.0800.0000.106F   ;1960  d_dal.sc   ; D=001F0000
U 106F. 0000.003C.3D80.3C00.0000.1077   ;1961  id[psl]_d   ; psl = 001F0000
U 1077. 0818.0038.0580.0800.0000.107B   ;1962  d_k[.1]   ; Clear the CES register
U 107B. 0D00.003C.0180.0800.0000.107F   ;1963  d_dal.sc   ; D = 00010000
U 107F. 0F00.003C.3180.3C00.0000.1083   ;1964  id[ces]_d,d_0   ; ces = 00010000
U 1083. 0000.003C.5D80.3C00.0000.1087   ;1965  id[acc.cs]_d   ; acc.cs = 0
U 1087. 0000.003C.3980.3C00.0000.108B   ;1966  id[sir]_d   ; sir = 0
U 108B. 0000.003C.2980.3C00.0000.1097   ;1967  id[clk.cs]_d   ; clk.cs = 0
U 1097. 0000.003C.4980.3C00.0000.1058   ;1968  id[tber0]_d   ; tber0 = 0
U 1058. 0000.003D.0180.0800.0000.120B   ;1969 =0 call[sbi.unjam]   ; Unjam the SBI
                ;1970  intrpt.strobe,   ; Strobe interrupts
U 1059. 0818.0038.C180.0800.4000.109F   ;1971      d_k[.ffff]   ; Setup to clear SBI error register and
U 109F. 0801.0028.6580.3C00.0000.10A7   ;1972  id[sbi.err]_d,d_not.d ; and fault register
U 10A7. 0F00.003C.6D80.3C00.0000.10AF   ;1973  id[fault]_d,d_0   ; ...
U 10AF. 0000.003C.6D80.3C00.0000.1109   ;1974  id[fault]_d   ; fault = 0
U 1109. 0000.003C.7580.3C00.0000.11DE   ;1975  id[maint]_d   ; MAINT = 0
U 11DE. 0000.003C.6580.3C00.0000.11DF   ;1976  id[sbi.err]_d   ; sbi error reg = 0
U 11DF. 0000.003C.7180.3C00.0000.11E0   ;1977  id[comp]_d   ; comp reg = 0
U 11E0. 0000.003C.E580.3C00.0000.11E1   ;1978  id[T9]_d   ; Clear code for subroutine START.TEST
U 11E1. 0001.003C.0180.09F0.0000.11E2   ;1979  rc[0e]_d   ;
U 11E2. 0001.003C.0180.0AF8.0000.11E3   ;1980  r[0f]_d   ; Clear expected trap mask
U 11E3. 0001.003C.0180.09F8.0000.105C   ;1981  rc[0f]_d   ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,1211  ;1982 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1983  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1984  .PAGE
            ;1985  .TOC "   Error Loop Routine"
            ;1986 ;**
            ;1987 ; FUNCTIONAL DESCRIPTION:
            ;1988 ;
            ;1989 ; This routine is called with the "ERLOOP" macro. The
            ;1990 ; routine calls the Micro Monitor to setup an error loop.
            ;1991 ;
            ;1992 ; CALLING CONSTRAINT:
            ;1993 ;
            ;1994 ; =0
            ;1995 ;
            ;1996 ; SIDE EFFECTS:
            ;1997 ;
            ;1998 ; D Reg - Destroyed
            ;1999 ;--
U 11E4, 0818,0038,0980,0800,0000,105E  ;2000 ERLOOP: d_k[.2],j/callcon
```

```
            ;2001  .PAGE
            ;2002  .TOC "   Error 1 Routine"
            ;2003 ;**
            ;2004 ; FUNCTIONAL DESCRIPTION:
            ;2005 ;
            ;2006 ; This routine is used to report an error to the user. This
            ;2007 ; routine is used when you do not wish to continue in the
            ;2008 ; same test after the call to the Monitor.
            ;2009 ;
            ;2010 ; CALLING CONSTRAINT:
            ;2011 ;
            ;2012 ; None
            ;2013 ;
            ;2014 ; INPUTS:
            ;2015 ;
            ;2016 ; rc[0f]<15:8> - Contain the subtest number
            ;2017 ;
            ;2018 ; OUTPUTS:
            ;2019 ;
            ;2020 ; D<15:8> - Subtest number
            ;2021 ; D<7:0> - Error 1 Monitor Code
            ;2022 ;--
U 11E5. 0810.0038.0180.0978.0000.11E6  ;2023 ERROR1: d_rc[0f]  ; Get the subtest number
U 11E6. 0819.0034.4D80.0800.0000.104E  ;2024  d_d.and.k[.ff00] ; ...
U 104E. 0819.0030.1180.0800.0000.105E  ;2025 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2026  .PAGE
            ;2027  .TOC "   ERROR1 WITH PARAMETER"
            ;2028 ;**
            ;2029 ; FUNCTIONAL DESCRIPTION:
            ;2030 ;
            ;2031 ; This subroutine takes  as parameter the number of arguments
            ;2032 ; to be printed to the console in the case of an error logout.
            ;2033 ;
            ;2034 ; CALLING CONSTRAINT:
            ;2035 ;
            ;2036 ; =0
            ;2037 ;INPUTS:
            ;2038 ;
            ;2039 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2040 ;--
            ;2041 =0
            ;2042 ERR1.ARG:
U 10B0. 0000.003D.0180.0800.0000.11ED  ;2043  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10B1. 0000.003C.0180.0800.0000.11E7  ;2044  NOP
U 11E7. 0000.003C.0180.0800.0000.11E5  ;2045  J/ERROR1  ; Go to ERROR1
U 11E8. 0000.003C.0180.0800.0000.11E9  ;2046  NOP
            ;2047 ;
            ;2048 ;
```

```
            ;2049  .PAGE
            ;2050  .TOC "   Error 2 Routine"
            ;2051 ;**
            ;2052 ; FUNCTIONAL DESCRIPTION:
            ;2053 ;
            ;2054 ; This routine is used  to report an error to the user. This
            ;2055 ; routine is used when you wish to continue in the same test
            ;2056 ;       after the call to the Monitor.
            ;2057 ;
            ;2058 ; CALLING CONSTRAINT:
            ;2059 ;
            ;2060 ; =0
            ;2061 ;
            ;2062 ; INPUTS:
            ;2063 ;
            ;2064 ; rc[0f]<15:8> - Contain the subtest number
            ;2065 ;
            ;2066 ; OUTPUTS:
            ;2067 ;
            ;2068 ; D<15:8> - Subtest number
            ;2069 ; D<7:0> - Error 2 Monitor Code
            ;2070 ;--
U 11E9, 0810,0038,0180,0978,0000,11EA  ;2071 ERROR2: d_rc[0f]  ; Get the subtest number
U 11EA, 0819,0034,4D80,0800,0000,105A  ;2072  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2073 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2074 ;
            ;2075 ; .PAGE
            ;2076  .TOC "   ERROR 2 WITH PARAMETER"
            ;2077 ;**
            ;2078 ; FUNCTIONAL DESCRIPTION:
            ;2079 ;
            ;2080 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2081 ; except that in this case after setting the number of arguments
            ;2082 ; too the console, control is transferred to routine ERROR2.
            ;2083 ;
            ;2084 ; INPUTS:
            ;2085 ;
            ;2086 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2087 ;
            ;2088 ;--
            ;2089 =0
            ;2090 ERR2.ARG:
U 10B2, 0000,003D,0180,0800,0000,11ED  ;2091  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10B3, 0000,003C,0180,0800,0000,11EB  ;2092  NOP
U 11EB, 0000,003C,0180,0800,0000,11E9  ;2093  J/ERROR2  ; Go to ERROR2
U 11EC, 0000,003C,0180,0800,0000,105E  ;2094  NOP
            ;2095 ;
```

```
            ;2096  .PAGE
            ;2097  .TOC "   Micro-Monitor Call"
            ;2098 ;*
            ;2099 ; FUNCTIONAL DESCRIPTION:
            ;2100 ;
            ;2101 ; This micro word calls the micro monitor.
            ;2102 ;
            ;2103 ; EXPLICIT INPUTS:
            ;2104 ;
            ;2105 ; D reg must contain valid monitor code.
            ;2106 ;--
            ;2107 105E:
            ;2108 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;2109  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2110  .PAGE
            ;2111  .TOC "   Reserved Control Store"
            ;2112 ;++
            ;2113 ; FUNCTIONAL DESCRIPTION:
            ;2114 ;
            ;2115 ; The following 16 locations must be reserved so the monitor
            ;2116 ; can use them to perform various functions that require
            ;2117 ; the execution of micro-code.
            ;2118 ;--
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2119 13F0: nop
U 13F1, 0000.003C.0180.0800.0000.13F2  ;2120 13F1: nop
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2121 13F2: nop
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2122 13F3: nop
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2123 13F4: nop
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2124 13F5: nop
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2125 13F6: nop
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2126 13F7: nop
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2127 13F8: nop
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2128 13F9: nop
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2129 13FA: nop
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2130 13FB: nop
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2131 13FC: nop
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2132 13FD: nop
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2133 13FE: nop
U 13FF, 0000.003C.0180.0800.0000.12AA  ;2134 13FF: nop
U 12AA, 0000.003C.0180.0800.0000.11ED  ;2135 12AA: nop ; This location is permanently blasted in the FPLA
            ;2136   ; to cause a micro ECO to location 12AA.
```

```
            ;2137  .PAGE
            ;2138  .TOC "   Set Argument Count"
            ;2139 ;++
            ;2140 ; FUNCTIONAL DESCRIPTION:
            ;2141 ;
            ;2142 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2143 ; to the console.
            ;2144 ;
            ;2145 ; CALLING CONSTRAINT:
            ;2146 ;
            ;2147 ; =0
            ;2148 ;
            ;2149 ; INPUTS:
            ;2150 ;
            ;2151 ; SC - Contains new value of argument count
            ;2152 ;
            ;2153 ; SIDE EFFECTS:
            ;2154 ;
            ;2155 ; Q is destroyed
            ;2156 ;--
U 11ED, 0010,0038,01C0,0978,0000,11EE  ;2157 SETRCF: q_rc[0f]  ; Get the argument count
U 11EE, 0019,2034,4DC0,0800,0000,11EF  ;2158  q_q.and.k[.ff00] ; ...
U 11EF, 0019,2032,1D80,09F8,0000,0001  ;2159  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2160  .PAGE
            ;2161  .TOC "   Increment Subtest Number"
            ;2162 ;++
            ;2163 ; FUNCTIONAL DESCRIPTION:
            ;2164 ;
            ;2165 ; This routine is used to increment the subtest number
            ;2166 ; in RC[0F]<15:8>.
            ;2167 ;
            ;2168 ; CALLING CONSTRAINT:
            ;2169 ;
            ;2170 ; =0
            ;2171 ;
            ;2172 ; SIDE EFFECTS:
            ;2173 ;
            ;2174 ; D register is destroyed
            ;2175 ;--
            ;2176 subtst:
U 11F0, 0810,0038,4D80,0978,0000,11F1  ;2177  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2178  rc[0f]_d+k[.ff00], ; Increment and return
U 11F1, 0019,4016,4D80,09F8,0000,0001  ;2179      word,return1 ; (Subtest number is negative)
```

```
            ;2180  .PAGE
            ;2181  .TOC " Diagnostic Specific Subroutines"
            ;2182  .TOC "   Select Controller"
            ;2183 ;++
            ;2184 ; FUNCTIONAL DESCRIPTION:
            ;2185 ;
            ;2186 ; This routine is used to put the memory system into the
            ;2187 ; specified configuration with respect to controller select.
            ;2188 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2189 ; a RETURN2 is made.
            ;2190 ;
            ;2191 ; CALLING CONSTRAINT:
            ;2192 ;
            ;2193 ; =00
            ;2194 ;
            ;2195 ; INPUTS:
            ;2196 ;
            ;2197 ; d - Contains value to write to bits<2:0> of Register A
            ;2198 ; rc[0e] - Address of Register A
            ;2199 ;
            ;2200 ; SIDE EFFECTS:
            ;2201 ;
            ;2202 ; sc,d,va are destroyed
            ;2203 ;--
            ;2204 SELECT.CNTRLR:
U 11F2, 0010.003B,0180.0970.0284.71F3  ;2205  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11F3, 0801.001C,0180.0800.0000.11F4  ;2206  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11F4, 0000.003C,0180.A800.0000.11F5  ;2207  cache.p_d[long]  ; Write the configuration Register
U 11F5, 0000.003C,0180.0800.0000.11F6  ;2208  NOP
U 11F6, 0000.003C,0180.0800.0000.11F7  ;2209  NOP
U 11F7, 0000.003C,0180.0800.0000.11F8  ;2210  NOP
U 11F8, 0000.003C,0180.0800.0000.11F9  ;2211  NOP
U 11F9, 0000.003C,0180.0800.0000.11FA  ;2212  NOP
U 11FA, 0000.003C,0180.0800.0000.11FB  ;2213  NOP
U 11FB, 0000.003C,0180.0800.0000.11FC  ;2214  NOP
U 11FC, 0000.003C,0180.0800.0000.11FD  ;2215  NOP
U 11FD, 0818.0038,5D80.0800.0000.11FE  ;2216  d_k[.7]   ; Get Misconfiguration bits
U 11FE, 0000.003C,61F8.0800.0084.71FF  ;2217  sc_k[.F],q_0  ; ...
U 11FF, 0D00.003C,0180.0800.0000.1200  ;2218  d_dal.sc  ; ... D = x38000
U 1200, 0000.003C,01E0.0800.0000.1201  ;2219  q_d   ; Save mask
U 1201, 0000.003C,0180.C800.0000.1202  ;2220  d[long]_cache.p  ; Read CNFG A Reg and
            ;2221  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1202, 001D.2034,0180.0800.0010.1203  ;2222   clk.ubcc ; ...
U 1203, 0000.013C,0180.0800.0000.10B4  ;2223  z?   ; Branch if no
U 10B4, 0000.003E,0180.0800.0000.0001  ;2224 =0 RETURN1   ; Bad configuration
U 10B5, 0000.003E,0180.0800.0000.0002  ;2225  RETURN2   ; Configuration ok
            ;2226
```

```
;2227  .PAGE
;2228  .TOC "   START TEST"
;2229 ;*************************************************************
;2230 ;**
;2231 ;
;2232 ; Functional Description:
;2233 ; ---------------------
;2234 ;
;2235 ;         This subroutine will be called by all tests that  check  the
;2236 ;    controllers,  or  those  tests  that  have  to switch between the
;2237 ;    controllers when executing.   Its  main  functions  are:   select
;2238 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2239 ;    test that called it, call out an error  if  neither  one  of  the
;2240 ;    controllers can be configured properly, if all the controllers on
;2241 ;    a MS780-E/H were configured and tested, it should select the next
;2242 ;    MS780-E/H on the SBI and repeat the above sequence.
;2243 ;    A test making use of this  subroutine  should  be  configured  as
;2244 ;    follows:
;2245 ;
;2246 ;
;2247 ;       Begin TEST.XX
;2248 ;       :
;2249 ;       :   Call NEWTST
;2250 ;       :   Call FIND.MS780E
;2251 ;       :   IF No MS780-E's on the SBI
;2252 ;       :     THEN
;2253 ;       :         Skip to End of TEST.XX
;2254 ;       :     ELSE
;2255 ;       :
;2256 ;       : RESTART.TEST.XX:
;2257 ;       :
;2258 ;       :   Call START TEST
;2259 ;       :   IF Return1 from START.TEST
;2260 ;       :     THEN
;2261 ;       :         Skip to End of TEST.XX
;2262 ;       :     ELSE.
;2263 ;       :
;2264 ;       ~
;2265 ;       :
;2266 ;       :
;2267 ;       :   Body of TEST.XX
;2268 ;       :
;2269 ;       ~
;2270 ;       :
;2271 ;       :   Jump RESTART.TEST.XX
;2272 ;       :
;2273 ;      End TEST.XX
;2274 ;
;2275 ;
;2276 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2277 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2278 ;    should be passed when the subroutine is called a second, third or
;2279 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2280 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2281 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2282 ;    Register 39 can be interpreted as follows:
;2283 ;
;2284 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2285 ;               called the first time.  When this code is encountered
;2286 ;               this subroutine will try to select the lower controller.
;2287 ;               If the lower controller is misconfigured, the pointer in
;2288 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2289 ;               is re-entered.
;2290 ;               If, however there is no misconfiguration, the value in
;2291 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2292 ;               calling test.
;2293 ;
;2294 ;     b01   -   This value signifies that an attempt at configuring the
;2295 ;               lower controller failed, and the subroutine will attempt
;2296 ;               to select the upper controller.
;2297 ;               If the upper controller is also misconfigured execution
;2298 ;               stops with a call to ERROR1.
;2299 ;               If there is no misconfiguration on this controller, then
;2300 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2301 ;               made to the calling test.
;2302 ;
;2303 ;     b10   -   This value signifies that the lower controller was selected
;2304 ;               previously and the test was executed once.  If the
;2305 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2306 ;               39 is loaded with b11 and an attempt is made to select the
;2307 ;               upper controller.
;2308 ;               If there is a misconfiguration on this controller, this
;2309 ;               subroutine is just re-entered.
;2310 ;               Otherwise, a Return2 will be made to the calling test.
;2311 ;
;2312 ;     b11   -   This code identifies the cases in which the test has
;2313 ;               been executed at least once (i.e., at least one of the
;2314 ;               controllers on the MS780-E unit under test could be
;2315 ;               configured properly).  When this code is detected the
;2316 ;               subroutine clears ID Reg 39 and makes a call to
;2317 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2318 ;               tested then the subroutine executes a Return1.
;2319 ;               Otherwise the subroutine is re-entered.
;2320 ;
;2321 ;
;2322 ; Calling Constraint: =00
;2323 ; ------------------
;2324 ;
;2325 ;
;2326 ; Inputs:
;2327 ; ------
;2328 ;
;2329 ; ID Register 39 must be cleared by NEWTST
;2330 ;
;2331 ;
;2332 ; Outputs:
;2333 ; -------
;2334 ;
;2335 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2336 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2337 ;         to be tested
            ;2338 ;
            ;2339 ; Side Effects:
            ;2340 ; ------------
            ;2341 ;
            ;2342 ; Contents of the following regiters will be destroyed:
            ;2343 ;
            ;2344 ; D, Q
            ;2345 ;
            ;2346 ;--
            ;2347 ;*****************************************************************
            ;2348 =0
            ;2349 START.TEST:
U 10B6, 0000,003D,0180,0800,0000,120C  ;2350  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 10B7, 0000,003C,0180,0800,0000,10B8  ;2351  NOP    ; ...tests that have more than one
            ;2352 =0     ; ...subtest.)
U 10B8, 0000,003D,0180,0800,0000,11E4  ;2353  ERLOOP    ; Set up the error loop for the
            ;2354       ; ...eventuality that the MS780-E/H
            ;2355       ; ...currently under test has no
            ;2356       ; ...Controllers
            ;2357 RE.ENTRY:    ; Re entry point for this routine
U 10B9, 0000,003C,E5F0,2C00,0000,1204  ;2358  Q_ID[39]    ; Get the code
U 1204, 0C00,003C,0180,0800,0000,1205  ;2359  D_Q    ; Put it in the D Register
U 1205, 0000,193C,0180,0800,0000,100C  ;2360  DI-0?    ; Branch on the code
U 100C, 0000,003C,0180,0800,0000,1008  ;2361 =1100 J/STRT.CASE00    ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1010  ;2362  J/STRT.CASE01    ; Code is 01
U 100E, 0000,003C,0180,0800,0000,120A  ;2363  J/STRT.CASE10    ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1017  ;2364  J/STRT.CASE11    ; Code is 11
            ;2365 ;
            ;2366 ;
            ;2367 ; At this point the code in ID[39] was 00
            ;2368 ;
            ;2369 ;
            ;2370 =00
            ;2371 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1024  ;2372  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2373  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,1206  ;2374   D_K[.1]    ; Set up the code for re entry to this
            ;2375       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2376  D_K[.2]    ; Set up the code for a next call to
            ;2377       ; ...START.TEST indicating that the
            ;2378       ; ...Lower Controller was configured
            ;2379       ; ...( 10 )
            ;2380  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2381   RETURN2    ; Let the test know that the Lower
            ;2382       ; ...has been configured
            ;2383 STRT.LOW.MIS:
            ;2384  ID[39]_D,    ; Save code in ID[39] signifing that
U 1206, 0000,003C,E580,3C00,0000,10B9  ;2385   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2386       ; ...and re enter this subroutine
            ;2387
            ;2388 ;
            ;2389 ;
            ;2390 ; At this point the code is 01
            ;2391 ;
```

```
            ;2392 ;
            ;2393 =00
            ;2394 STRT.CASE01:
U 1010, 0000,003D,0180,080D,0000,1028  ;2395  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2396  J/STRT.UPR.MIS.   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,1207  ;2397   D_0    ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2398  D_K[.3]    ; Upper controller was configured
            ;2399       ; ...thus the code must be changed
            ;2400       ; ...accordingly ( 11 )
            ;2401  ID[39]_D,    ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2402   RETURN2    ; Let the test know that the Upper
            ;2403       ; ...Controller has been configured
            ;2404 STRT.UPR.MIS:
U 1207, 0000,003C,E580,3C00,0000,1208  ;2405  ID[39]_D    ; Save the Code ( 00 )
U 1208, 0018,0038,0D80,09E8,0000,1209  ;2406  RC[0D]_K[.3]    ; Set up a code for the Fail Chain
U 1209, 0000,003D,0980,0800,0084,70B0  ;2407  ERROR1[.2]    ; Flag the error.
            ;2408 ;
            ;2409 ;
            ;2410 ; At this point the code is 10
            ;2411 ;
            ;2412 ;
            ;2413 STRT.CASE10:
U 120A, 0818,0038,0D80,0800,0000,1014  ;2414  D_K[.3]    ; Set the code to 11
            ;2415
            ;2416
            ;2417 =00 ID[39]_D,    ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1028  ;2418   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,10B9  ;2419  J/RE.ENTRY    ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2420  RETURN2     ; Upper Controller configured
            ;2421       ; ...let the test know it
            ;2422 ;
            ;2423 ;
            ;2424 ; At this point the code is 11
            ;2425 ;
            ;2426 ;
            ;2427 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1020  ;2428  D_0     ; Clear the code
            ;2429 =00 ID[39]_D,    ; Save the code
U 1020, 0000,003D,E580,3C00,0000,1275  ;2430   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2431       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,10B9  ;2432  J/RE.ENTRY    ; Found another MS780-E/H.   Go and
            ;2433       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2434  RETURN1     ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1024  ;2435  NOP
```

```
;2436  .PAGE
;2437  .TOC "   SELECT LOWER CONTROLLER"
;2438 ;*************************************************************
;2439 ;**
;2440 ;
;2441 ; Functional Description:
;2442 ; ----------------------
;2443 ;
;2444 ; This subroutine can be called to select the lower
;2445 ; Controller on a MS780-E/H.
;2446 ; If the Lower controller is misconfigured then a
;2447 ; RETURN1 will be made.  Otherwise a RETURN2
;2448 ;
;2449 ; Calling Constraint: =00
;2450 ; ------------------
;2451 ;
;2452 ;
;2453 ; Inputs:
;2454 ; ------
;2455 ;
;2456 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2457 ;
;2458 ; Outputs:
;2459 ; -------
;2460 ;
;2461 ; RC[0D] will have the address of CNFG Register C
;2462 ;        ( 2000x008 )
;2463 ;
;2464 ; Side Effects:
;2465 ; ------------
;2466 ;
;2467 ; Contents of the following registers will lost:
;2468 ;
;2469 ; D
;2470 ;
;2471 ; The Lower controller on the MS780-E/H will be configured
;2472 ; when the subroutine is exitted with a RETURN2
;2473 ;--
;2474 ;*************************************************************
;2475 =00
;2476 SELECT.LOWER:
;2477  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,11F2  ;2478   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2479  RETURN1    ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2480  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2481  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2482   RETURN2   ; Let caller know that the controller
;2483       ; ...was configured properly
```

```
;2484  .PAGE
;2485  .TOC "   SELECT UPPER CONTROLLER"
;2486 ;*************************************************************
;2487 ;**
;2488 ;
;2489 ; Functional Description:
;2490 ; ----------------------
;2491 ;
;2492 ; This subroutine can be called to select the upper
;2493 ; Controller on a MS780-E/H.
;2494 ; If the Upper controller is misconfigured then a
;2495 ; RETURN1 will be made.  Otherwise a RETURN2
;2496 ;
;2497 ; Calling Constraint: =00
;2498 ; ------------------
;2499 ;
;2500 ;
;2501 ; Inputs:
;2502 ; ------
;2503 ;
;2504 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2505 ;
;2506 ; Outputs:
;2507 ; -------
;2508 ;
;2509 ; RC[0D] will have the address of CNFG Register D
;2510 ;        ( 2000x010 )
;2511 ;
;2512 ; Side Effects:
;2513 ; ------------
;2514 ;
;2515 ; Contents of the following registers will lost:
;2516 ;
;2517 ; D
;2518 ;
;2519 ; The Upper controller on the MS780-E/H will be configured
;2520 ; when the subroutine is exitted with a RETURN2
;2521 ;--
;2522 ;*************************************************************
;2523 =00
;2524 SELECT.UPPER:
;2525  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,11F2  ;2526   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2527  RETURN1     ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2528  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2529  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2530   RETURN2    ; Let caller know that the controller
;2531       ; ...was configured properly
;2532  .TOC "    SBI Unjam Routine"
;2533 ;++
;2534 ; FUNCTIONAL DESCRIPTION:
;2535 ;
;2536 ; This routine performs an SBI UNJAM sequence. This is done by
;2537 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
;2538 ; and finally HOLD for the last 16 cycles.
```

```
            ;2539 ;
            ;2540 ; CALLING CONSTRAINT:
            ;2541 ;
            ;2542 ; =0
            ;2543 ;
            ;2544 ; SIDE EFFECTS:
            ;2545 ;
            ;2546 ; D Reg destroyed
            ;2547 ;--
            ;2548   ;
            ;2549   ; assert hold for 16 cycles
            ;2550   ;
            ;2551 SBI.UNJAM:
            ;2552  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 120B, 0818,0038,6580,8000,0010,10BA  ;2553      clk.ubcc  ; set micro cc's for next cycle
            ;2554 =0
            ;2555 SBI.UNJAM.1:
            ;2556  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2557       clk.ubcc,z?, ; ...
U 10BA, 0819,0100,0580,8000,0010,10BA  ;2558      j/sbi.unjam.1 ; test for completion
            ;2559    ;
            ;2560    ; assert hold and unjam for 16 cycles
            ;2561    ;
            ;2562  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10BB, 0818,0038,6580,8800,0010,10BC  ;2563      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2564 =0
            ;2565 SBI.UNJAM.2:
            ;2566  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2567       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10BC, 0819,0100,0580,8800,0010,10BC  ;2568      z?,j/sbi.unjam.2 ; ...
            ;2569    ;
            ;2570    ; assert hold for 16 cycles
            ;2571    ;
            ;2572  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10BD, 0818,0038,6580,8000,0010,10BE  ;2573      clk.ubcc  ; and set cc's for next cycle
            ;2574 =0
            ;2575 SBI.UNJAM.3:
            ;2576  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2577       clk.ubcc,z?, ; ...
U 10BE, 0819,0100,0580,8000,0010,10BE  ;2578      j/sbi.unjam.3 ; ...
U 10BF, 0000,003E,0180,0800,0000,0001  ;2579  return1   ; exit
```

```
            ;2580  .PAGE
            ;2581  .TOC "   RESET SUBTEST NUMBER"
            ;2582 ;**
            ;2583 ; FUNCTIONAL DESCRIPTION:
            ;2584 ;
            ;2585 ; Since some  of the tests will have to be restarted when two
            ;2586 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2587 ; the subtest counter to zero ( only for thoes tests that have
            ;2588 ; more than one subtest). This subroutine may also be necessary
            ;2589 ; when a test is being loop on.
            ;2590 ;
            ;2591 ; CALLING CONSTRAINT:
            ;2592 ;
            ;2593 ; =0
            ;2594 ; SIDE EFFECTS:
            ;2595 ;
            ;2596 ; D is destroyed
            ;2597 ;
            ;2598 ;--
            ;2599 RESET.SUBTST:
            ;2600  RC[0F] 0,   ; Reset subtest number
U 120C. 0003,003E,0180,09F8,0000,0001  ;2601   RETURN1   ; ...and return
```

```
            ;2602  .PAGE
            ;2603  .TOC "   Initialize the SBI"
            ;2604 ;++
            ;2605 ; FUNCTIONAL DESCRIPTION:
            ;2606 ;
            ;2607 ; This routine performs the following functions:
            ;2608 ;
            ;2609 ;  Executes an SBI Unjam
            ;2610 ;  Clears the SBI Fault Latch
            ;2611 ;  Clears the SBI Error Register
            ;2612 ;
            ;2613 ; CALLING CONSTRAINT:
            ;2614 ;
            ;2615 ; =0
            ;2616 ;
            ;2617 ; SIDE EFFECTS:
            ;2618 ;
            ;2619 ; D & Q Register destroyed
            ;2620 ;--
            ;2621 =0
            ;2622 SBI.INIT:
U 10C0. 0000,003D,0180,0800,0000,120B  ;2623  call[sbi.unjam]  ; Unjam the SBI
U 10C1. 0818,0038,C180,0800,0000,120D  ;2624  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 120D. 0801,0028,6580,3C00,0000,120E  ;2625  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 120E. 0F00,003C,6D80,3C00,0000,120F  ;2626  id[fault]_d,d_0
U 120F. 0000,003C,6D80,3C00,0000,1210  ;2627  id[fault]_d
U 1210. 0000,003E,6580,3C00,0000,0001  ;2628  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2629  .PAGE
            ;2630  .TOC "   Disable Cache Routine"
            ;2631 ;**
            ;2632 ; FUNCTIONAL DESCRIPTION:
            ;2633 ;
            ;2634 ; This routine disables cache hits by setting bits <16:15>
            ;2635 ; in the maintenance register.
            ;2636 ;
            ;2637 ; CALLING SEQUENCE:
            ;2638 ;
            ;2639 ; =0
            ;2640 ;
            ;2641 ; SIDE EFFECTS:
            ;2642 ;
            ;2643 ; D & Q Registers destroyed
            ;2644 ;--
            ;2645 DISABLE.CACHE:
U 1211, 0818,0038,4580,0800,0000,1212  ;2646  d_k[.8000]  ; ... and seed constant
U 1212, 0500,003C,0180,0800,0000,1213  ;2647  d_d.left  ; Generate final constant
U 1213, 0819,0030,4580,0800,0000,1214  ;2648  d_d.or.k[.8000]  ; ... = 00018000
U 1214, 0000,003E,7580,3C00,0000,0001  ;2649  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2650  .PAGE
            ;2651  .TOC "   Enable Cache"
            ;2652 ;**
            ;2653 ; FUNCTIONAL DESCRIPTION:
            ;2654 ;
            ;2655 ; This routine enables cache group 0 by clearing the force
            ;2656 ; miss bit in the maintenance register.
            ;2657 ;
            ;2658 ; CALLING CONSTRAINT:
            ;2659 ;
            ;2660 ; =0
            ;2661 ;
            ;2662 ; SIDE EFFECTS:
            ;2663 ;
            ;2664 ; D, Q AND SC Registers destroyed
            ;2665 ;--
            ;2666 ENABLE.CACHE:
U 1215. 0000.003C,75F0.2C00.0000.1216  ;2667  q_id[maint]  ; Get current maintenance register
U 1216. 0000.003C,6580.0800.0084.7217  ;2668  sc_k[.10]  ; Setup to clear bit 16
U 1217. 0801.2034,0180.0800.0000.1218  ;2669  d_q.and.mask  ; Clear bit 16
U 1218. 0000.003E,7580.3C00.0000.0001  ;2670  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2671  .PAGE
            ;2672  .TOC "   Wait Loop Routine"
            ;2673 ;++
            ;2674 ; FUNCTIONAL DESCRIPTION:
            ;2675 ;
            ;2676 ; As the name implies,  this subroutine is used to set up a wait
            ;2677 ; loop for a specified  number of cycles.  This may be necessary
            ;2678 ; when the micro-program has to wait for another event to finish.
            ;2679 ; When the subroutine is entered Register D should be holding the
            ;2680 ; desired numbered of cycles.
            ;2681 ;
            ;2682 ; CALLING CONSTRAINT:
            ;2683 ;
            ;2684 ; =0
            ;2685 ;
            ;2686 ; INPUTS:
            ;2687 ;
            ;2688 ; Q - Contains number of cycles to wait
            ;2689 ; alu.z condition code must not be set
            ;2690 ;
            ;2691 ; SIDE EFFECTS:
            ;2692 ;
            ;2693 ; Q is destroyed
            ;2694 ;--
            ;2695 WAIT.LOOP:
U 1219, 0018,0038,6180,0800,0010,10C2  ;2696  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2697      ; ...is not set
            ;2698 =0
            ;2699 W.LOOP:
            ;2700  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10C2, 0819,0100,0580,0800,0010,10C2  ;2701       j/W.LOOP
U 10C3, 0000,003E,0180,0800,0000,0001  ;2702  return1    ; exit
            ;2703
```

```
            ;2704  .PAGE
            ;2705  .TOC "   Check for Interrupt Routine"
            ;2706 ;++
            ;2707 ; FUNCTIONAL DESCRIPTION:
            ;2708 ;
            ;2709 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2710 ; If an interrupt is active, the following registers are setup:
            ;2711 ;  RC[8] - Gets the VECTOR register
            ;2712 ;  RC[7] - Gets the SBI ERROR register
            ;2713 ;  RC[6] - Gets the FAULT register
            ;2714 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2715 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2716 ;
            ;2717 ; CALLING CONSTRAINT:
            ;2718 ;
            ;2719 ; =00
            ;2720 ;
            ;2721 ; SIDE EFFECTS:
            ;2722 ;
            ;2723 ; D & Q are destroyed
            ;2724 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2725 ;--
            ;2726 CHECK_INTERRUPT:
            ;2727   ;
            ;2728   ; First, enable the fault and RDS/CRD interrupts
            ;2729   ;
U 121A, 0818,0038,4580,0800,0000,121B  ;2730  d_k[.8000]  ; Get data to set interrupt enable
            ;2731  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 121B, 0500,003C,6580,3C00,0000,121C  ;2732      d_d.left
U 121C, 0100,003C,0180,0800,0000,121D  ;2733  d_d.left2  ; D = 40000
U 121D, 0000,003C,6D80,3C00,0000,121E  ;2734  id[fault]_d  ; Set fault interrupt enable
            ;2735  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 121E, 0F03,003C,0180,09A8,4000,121F  ;2736      rc[5]_0  ; and clear interrupt occurred flag
U 121F, 0000,003C,3D80,3C00,0000,1220  ;2737  id[psl]_d  ; Clear the PSL
U 1220, 0000,003C,6580,3C00,0000,1221  ;2738  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1221, 0000,003C,6D80,3C00,0000,1222  ;2739  id[fault]_d  ; Clear FAULT Interrupt Enable
U 1222, 0000,0E3C,0180,0800,0000,1004  ;2740  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2741 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2742  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2743  return1   ; No interrupt
            ;2744 =111
            ;2745 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,1223  ;2746  q_id[vector]  ; Get contents of ID vector register
U 1223, 0001,203C,65F0,2DC0,0000,1224  ;2747  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 1224, 0001,203C,6DF0,2DB8,0000,1225  ;2748  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 1225, 0001,203C,0180,09B0,0000,1226  ;2749  rc[6]_q   ; Save falut reg
U 1226, 0018,003A,0580,09A8,0000,0002  ;2750  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2751  .PAGE
            ;2752  .TOC "   CHECK UTRAP"
            ;2753 ;++
            ;2754 ;FUNCTIONAL DESCRIPTION:
            ;2755 ;
            ;2756 ; This subroutine is used to check wether a Utrap occured.
            ;2757 ; It takes the contents of the Save Utrap flags register
            ;2758 ; R[0E], and compares them to the contrnts of the Utrap
            ;2759 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2760 ; according to the results of the test  (i.e., contents of
            ;2761 ; these registers are equal or not).
            ;2762 ; CALLING CONSTRAINT:
            ;2763 ;
            ;2764 ; =00
            ;2765 ;
            ;2766 ; INPUTS:
            ;2767 ;
            ;2768 ; R[0E]- Saved Utrap Mask
            ;2769 ; R[0F]- Expected Utrap Mask
            ;2770 ;
            ;2771 ;--
            ;2772 CHECK_UTRAP:
U 1227, 0800,003C,0180,0A70,0000,1228  ;2773  D_R[0E]   ; Reg D is loaded with the trap mask
U 1228, 0001,003C,0180,09C0,0000,1229  ;2774  RC[08]_D  ; Save expected Utrap flag for error logout
U 1229, 0800,003C,0180,0A78,0000,122A  ;2775  D_R[0F]   ; Get recieved Utrap flag to D reg
U 122A, 0001,003C,0180,09B8,0000,122B  ;2776  RC[07]_D  ; Save in RC[07]
            ;2777  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 122B, 000D,0020,0180,0A70,0010,122C  ;2778         CLK.UBCC
U 122C, 0003,013C,0180,0AF8,0000,10C4  ;2779  Z?,R[0F]_0  ; Branch if no traps
U 10C4, 0000,003E,0180,0800,0000,0002  ;2780 =0 RETURN2   ; Utrap occured
U 10C5, 0000,003E,0180,0800,0000,0001  ;2781  RETURN1    ; No Utrap  return
            ;2782 ;
```

```
            ;2783  .PAGE
            ;2784  .TOC "   CHECK SBI.ERR"
            ;2785 ;**
            ;2786 ; FUNCTIONAL DESCRIPTION:
            ;2787 ;
            ;2788 ; This subroutine can be used to see whether the SBI.ERR Register
            ;2789 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;2790 ; routine will have to pass to this subroutine in the SC register
            ;2791 ; the number of the error bit to checked.
            ;2792 ;
            ;2793 ; CALLING CONSTRAINT:
            ;2794 ;
            ;2795 ; =00
            ;2796 ;
            ;2797 ; INPUTS:
            ;2798 ;
            ;2799 ; SC with the number of the error bit to be checked
            ;2800 ;
            ;2801 ; OUTPUTS:
            ;2802 ;
            ;2803 ; RC[08]-Contains the SBI.ERROR Register
            ;2804 ;--
            ;2805 CHECK_SBI_ERR:
U 122D, 0000,003C,65F0,2C00,0000,122E   ;2806  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 122E, 0001,203C,0180,09C0,0000,122F   ;2807  RC[08]_Q  ; Save for error logout
            ;2808  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 122F, 0001,2024,0180,0800,0010,1230   ;2809   CLK.UBCC ; ...
U 1230, 0000,013C,0180,0800,0000,10C6   ;2810  Z?   ; Check condition codes
U 10C6, 0000,003E,0180,0800,0000,0002   ;2811 =0 RETURN2   ; Bit is set
U 10C7, 0000,003E,0180,0800,0000,0001   ;2812  RETURN1   ; Bit is not set
```

```
            ;2813  .PAGE
            ;2814  .TOC "   Set Trap Mask"
            ;2815 ;**
            ;2816 ; FUNCTIONAL DESCRIPTION:
            ;2817 ;
            ;2818 ; This routine is used to set a bit in the Micro Trap Mask
            ;2819 ; R[0F].
            ;2820 ;
            ;2821 ; CALLING CONSTRAINT:
            ;2822 ;
            ;2823 ; =0
            ;2824 ;
            ;2825 ; INPUTS:
            ;2826 ;
            ;2827 ; SC - Contains bit number to set.
            ;2828 ;
            ;2829 ; OUTPUTS:
            ;2830 ;
            ;2831 ; rc[0E] - Contains the current trap mask
            ;2832 ;
            ;2833 ; SIDE EFFECTS:
            ;2834 ;
            ;2835 ; d regsiter is destroyed
            ;2836 ;--
            ;2837 SET_TRAP_MASK:
U 1231, 0800.003C,0180.0A78.0000.1232  ;2838  d_r[0f]
U 1232, 0801.001C,0180.0AF8.0000.1233  ;2839  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1233, 0001.003E,0180.0AF0.0000.0001  ;2840  r[0E]_d,return1   ; Save mask and return
```

```
            ;2841  .PAGE
            ;2842  .TOC "   FIND MS780-E MEMORY"
            ;2843 ;++
            ;2844 ; FUNCTIONAL DESCRIPTION:
            ;2845 ;
            ;2846 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2847 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2848 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2849 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2850 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2851 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2852 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2853 ;     (zero).
            ;2854 ;     All the other MS780-E/H's found will have their starting  address
            ;2855 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2856 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2857 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2858 ;     current MS780-E unit under test.
            ;2859 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2860 ;     set to maximum.
            ;2861 ;
            ;2862 ; CALLING CONSTRAINT:
            ;2863 ;
            ;2864 ; =00
            ;2865 ;
            ;2866 ; OUTPUTS:
            ;2867 ;
            ;2868 ; rc[0e] - Contains address of Configuration Register
            ;2869 ; r[0]  - Contains TR level
            ;2870 ;
            ;2871 ; SIDE EFFECTS:
            ;2872 ;
            ;2873 ; D register is destroyed.
            ;2874 ;--
            ;2875 =0
            ;2876 FIND.MS780E:
U 10C8, 0000,003D,0180,0800,0000,10C0  ;2877  call[sbi.init]  ; Make sure SBI is enabled
U 10C9, 0003,003C,0180,0988,0000,1234  ;2878  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1234, 0818,0038,1980,0800,0000,1235  ;2879  d_k[zero]  ; Clear MS780-E/H counter
U 1235, 0000,003C,F980,3C00,0000,1236  ;2880  id[3e]_d  ; ...
U 1236, 0018,0038,0580,0A80,0000,1237  ;2881  r[0]_k[.1]  ; Init TR counter
U 1237, 0F03,003C,0180,09F0,0000,10CA  ;2882  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2883      ; ...CNFG A Reg
            ;2884 =0
            ;2885 FIND.MEM.LOOP:
U 10CA, 0800,003D,0180,0A00,0000,125D  ;2886  d_r[0],call[generate.io]; Generate address of TR level
U 10CB, 0001,003C,0180,09F0,0200,10CC  ;2887  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10CC, 0000,003D,8980,0800,0084,7231  ;2888 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2889  d[long]_cache.p, ; Read the specified tr level
U 10CD, 0000,003C,0180,C970,0000,1238  ;2890    ic_rc[0e]  ; ...
U 1238, 0000,003C,0180,0A78,0010,1239  ;2891  alu_r[0f],clk.ubcc ; Check for no response
U 1239, 0001,013C,0180,0880,0082,10CE  ;2892  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10CE, 0000,003C,0180,0800,0000,123A  ;2893 =0 j/check.adpt.code ; Check for a memory
U 10CF, 0000,003C,0180,0800,0000,1259  ;2894  j/find.mem.lp1  ; Try next tr
            ;2895 CHECK.ADPT.CODE:
```

```
U 123A, 0000,003C,1D80,0800,1404,723B  ;2896  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 123B, 0000,163C,0180,0800,0000,1018  ;2897  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,723C  ;2898 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,723D  ;2899  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,1259  ;2900  j/find.mem.lp1  ; Not a memory
U 101B, 0000,003C,0180,0800,0000,1259  ;2901  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,7242  ;2902  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,1259  ;2903  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,1246  ;2904  rc[0e]_ic,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,1246  ;2905  rc[0e]_ic,j/found.ms780e ; MS780-E
        ;2906   ;
        ;2907   ; Found an MS780-A or -C. Set the starting address
        ;2908   ; to maximum.
        ;2909   ;
U 123C, 0818,0038,5D80,0800,0000,123E  ;2910 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 123D, 0818,0038,0580,0800,0000,123E  ;2911 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
        ;2912 MS780.COMMON:
U 123E, 0819,0030,7180,0800,0000,123F  ;2913  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 123F, 0000,003C,01F8,0803,0080,D240  ;2914  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1240, 0D00,003C,0180,0800,0000,1241  ;2915  d_dal.sc  ; Put data in bits<29:14>
        ;2916  cache.p_d[long], ; Set the starting address to maximum
U 1241, 0000,003C,0180,A800,0000,1259  ;2917       j/find.mem.lp1 ; and continue TR scan
        ;2918   ;
        ;2919   ; Found an MA780. Set the starting address to maximum
        ;2920   ;
        ;2921 MA780.MAX:
U 1242, 0818,0038,39F8,0803,0000,1243  ;2922  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1243, 0D00,003C,0180,0803,0000,1244  ;2923  d_dal.sc,va_va+4 ; Put in proper position
U 1244, 0000,003C,0180,0803,0000,1245  ;2924  va_va+4   ; Point to Port Invalidate Ctrl Register
        ;2925  cache.p_d[long], ; Set starting address to maximum
U 1245, 0000,003C,0180,A800,0000,1259  ;2926       j/find.mem.lp1
        ;2927 ;
        ;2928 ; Found an MS780E
        ;2929 ;
        ;2930 FOUND.MS780E:
U 1246, 0000,003C,F9F0,2C00,0000,1247  ;2931  q_id[3e]  ; Set up to see how many MS780-E's
        ;2932  alu_q.xor.k[.3], ; Are there Maximum units?
U 1247, 0019,2020,0D80,0800,0010,1248  ;2933   clk.ubcc ; Check condition codes
U 1248, 0000,013C,0180,0800,0000,10D0  ;2934  z?   ; Are we at maximum units for system
U 10D0, 0000,003C,0180,0800,0000,1249  ;2935 =0 J/ms780e.tst  ; No go find how many units ther are
U 10D1, 0818,0038,C180,0800,0000,10D2  ;2936  d_k[.ffff]  ; Yes set address to maximum
U 10D2, 0000,003D,0180,0800,0000,1261  ;2937 =0 call[set.start.addr] ; .....
U 10D3, 0000,003C,0180,0800,0000,1259  ;2938  j/find.mem.lp1  ; Go check to see if we are done
        ;2939 ;
        ;2940 MS780E.TST:
        ;2941  alu_rc[01],  ; Check "Found MS780E Flag"
U 1249, 0010,0038,0180,0908,0010,124A  ;2942   clk.ubcc ; Check condition codes
U 124A, 0810,0138,0180,0970,0000,10D4  ;2943  z?,d_rc[0e]  ; Is this unit the first MS780E ?
        ;2944 =0 j/not.first.ms780e, ; No
U 10D4, 0818,0038,C180,0800,0000,10D8  ;2945   d_k[.ffff] ; Set starting address
U 10D5, 0001,003C,0180,0988,0000,124B  ;2946  rc[01]_d  ; Yes put base address in rc[01]
U 124B, 0818,0038,7980,0800,0000,124C  ;2947  d_k[.30]  ; Set up SC to point to first unit
U 124C, 0019,0014,F580,0800,0082,124D  ;2948  alu_d+k[.a],sc_alu ; ...
U 124D, 0810,0038,0180,0908,0000,124E  ;2949  d_rc[01]  ; Put base address of unit in D
U 124E, 0000,003C,0180,3400,0000,124F  ;2950  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 124F, 0F00.003C.0180.0800.0000.10D6  ;2951  d_0   ; Prepare to set starting address to Zero
U 10D6, 0000.003D.0180.0800.0000.1261  ;2952 =0 call[set.start.addr] ; Go do it
          ;2953  j/find.mem.lp1,   ; Check to see if we are done
U 10D7, 0003.003C.0180.09E8.0000.1259  ;2954   rc[0d]_0 ; ....
          ;2955 =0
          ;2956 NOT.FIRST.MS780E:
U 10D8, 0000.003D.0180.0800.0000.1261  ;2957  call[set.start.addr] ; Go set starting address to maximum
U 10D9, 0000.003C.F9F0.2C00.0000.1250  ;2958  q_id[3e]  ; Get the number of MS780-Es found
U 1250, 0819.2014.0580.0800.0000.1251  ;2959  d_q+k[.1]  ; Increment this counter
U 1251, 0000.003C.F980.3C00.0000.1252  ;2960  id[3e]_d  ; Save the counter
U 1252, 0018.0038.11C0.0800.0000.1253  ;2961  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2962     ; ...were the I/O base addresses will be stored
U 1253, 081D.2000.7980.0800.0000.1254  ;2963  d_q-d,k[.30]  ; ... adjust pointer
U 1254, 0819.0014.7980.0800.0000.1255  ;2964  d_d+k[.30]  ; Point the SC to the ID register
U 1255, 0000.003C.F580.0800.0000.1256  ;2965  k[.a]   ; ...to receive I/O base address
U 1256, 0019.0014.F580.0800.0082.1257  ;2966  alu_d+k[.a],sc_alu ; ...
U 1257, 0810.0038.0180.0970.0000.1258  ;2967  d_rc[0e]  ; Get base address to d
U 1258, 0000.003C.0180.3400.0000.1259  ;2968  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2969   ;
          ;2970   ; Try next tr
          ;2971   ;
          ;2972 FIND.MEM.LP1:
U 1259, 0818.0014.0580.0A80.0000.125A  ;2973  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2974  alu_d.and.k[.f],
U 125A, 0019.0034.6180.0800.0010.125B  ;2975       clk.ubcc  ; Check if all done
U 125B, 0000.013C.0180.0800.0000.10DA  ;2976  z?   ; Branch if yes
U 10DA, 0000.003C.0180.0800.0000.10CA  ;2977 =0 j/find.mem.loop  ; Try next TR
U 10DB, 0810.0038.0180.0908.0010.125C  ;2978  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 125C, 0000.013C.0180.0800.0000.10DC  ;2979  z?   ; Check condition codes
U 10DC, 0001.003E.0180.09F0.0000.0002  ;2980 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10DD, 0000.003E.0180.0800.0000.0001  ;2981  return1   ; No MS780E found
          ;2982
```

```
            ;2983   .PAGE
            ;2984   .TOC "   Generate IO Address"
            ;2985  ;++
            ;2986  ; FUNCTIONAL DESCRIPTION:
            ;2987  ;
            ;2988  ; This routine is used to generate an SBI Configuration Register
            ;2989  ; address from a TR level.
            ;2990  ;
            ;2991  ; CALLING CONSTRAINT:
            ;2992  ;
            ;2993  ; =0
            ;2994  ;
            ;2995  ; INPUTS:
            ;2996  ;
            ;2997  ; D - Contains TR level
            ;2998  ;
            ;2999  ; OUTPUTS:
            ;3000  ;
            ;3001  ; D - Contains SBI IO address
            ;3002  ;--
            ;3003  GENERATE.IO:
U 125D. 0000.003C.89F8.0800.0084.725E  ;3004  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 125E. 0D00.003C.8D80.0800.0084.725F  ;3005  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 125F. 0000.003C.0980.0800.0084.B260  ;3006  sc_sc-k[.2]  ; insert bit 29
U 1260. 0801.001E.0180.0800.0000.0001  ;3007  d_d.ornot.mask,return1 ; and return
```

```
            ;3008  .PAGE
            ;3009  .TOC "   Set Starting Address"
            ;3010 ;**
            ;3011 ; FUNCTIONAL DESCRIPTION:
            ;3012 ;
            ;3013 ; This routine is used to set the starting address of the
            ;3014 ; memory to the specified value.
            ;3015 ;
            ;3016 ; CALLING CONSTRAINT:
            ;3017 ;
            ;3018 ; =0
            ;3019 ;
            ;3020 ; INPUTS:
            ;3021 ;
            ;3022 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3023 ;     (B Register Image divided by 2**16)
            ;3024 ; rc[0e] - Contains Address of Register A
            ;3025 ;
            ;3026 ; OUTPUTS:
            ;3027 ;
            ;3028 ; d - Contains aligned starting address (B Register Image)
            ;3029 ;
            ;3030 ; SIDE EFFECTS:
            ;3031 ;
            ;3032 ; d,q,va, and sc are destroyed
            ;3033 ;--
            ;3034 SET.START.ADDR:
U 1261. 0010.0038.6580.0970.0284.7262  ;3035  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1262. 0000.003C.01F8.0803.0000.1263  ;3036  va_va+4,q_0  ; Set address to Register B
            ;3037  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1263. 0D00.003C.0980.0800.0084.B264  ;3038       sc_sc-k[.2]  ; Setup to insert bit 14
U 1264. 0801.001C.0180.0800.0000.1265  ;3039  d_d.ornot.mask  ; Insert Write Enable bit
U 1265. 0000.003E.0180.A800.0000.0001  ;3040  cache.p_d[long],return1 ; Set the starting address and return
            ;3041
            ;3042
```

```
            ;3043  .PAGE
            ;3044  .TOC "   FIND CPU ID"
            ;3045 ;*****************************************************
            ;3046 ;+*
            ;3047 ;
            ;3048 ; Functional Description:
            ;3049 ; ----------------------
            ;3050 ;
            ;3051 ; This subroutine finds the CPU ID by using the SILO.
            ;3052 ;
            ;3053 ;
            ;3054 ; Calling Constraint: =0
            ;3055 ; ------------------
            ;3056 ;
            ;3057 ;
            ;3058 ; Inputs:
            ;3059 ; ------
            ;3060 ;
            ;3061 ; NONE
            ;3062 ;
            ;3063 ;
            ;3064 ; Outputs:
            ;3065 ; -------
            ;3066 ;
            ;3067 ; The CPU ID is returned in the D Reg
            ;3068 ;
            ;3069 ;
            ;3070 ; Side Effects:
            ;3071 ; ------------
            ;3072 ;
            ;3073 ; The contents of the following registers will be lost:
            ;3074 ;
            ;3075 ; D, VA, ID[COMP], Q, SC
            ;3076 ;
            ;3077 ;--
            ;3078 ;*****************************************************
            ;3079 FIND.CPU.ID:
U 1266, 0003.003C.0180.0800.0200.1267  ;3080  VA_ALU,ALU_0(A)    ; FIND THE CPU ID
U 1267, 0818.0038.7580.0800.0000.126B  ;3081  D_K[.20]    ; Set up SILO Comparator to lock
U 1268, 0B00.003C.0180.0800.0000.1269  ;3082  D_D.SWAP    ; ...
U 1269, 0000.003C.7180.3C00.0000.126A  ;3083  ID[COMP]_D    ; START THE SILO COUNTER
U 126A, 0000.003C.0180.C800.0000.10DE  ;3084  D[LONG]_CACHE.P    ; ...
            ;3085 =0 D_K[.10],    ; Prepare to wait 16 cycles.
U 10DE, 0818.0039.6580.0800.0000.1219  ;3086    CALL[WAIT.LOOP]  ; Wait for the SILO to fill
            ;3087 KEEPLOOKING:
U 10DF, 0000.003C.61F0.2C00.0000.126B  ;3088  Q_ID[SILO]    ; SEARCH SILO FOR FIRST NON ZERO ENTRY
U 126B, 0C01.203C.7D80.0800.0010.126C  ;3089  ALU_Q,CLK.UBCC,K[.18],D_Q ;
U 126C, 001B.0100.7D80.0800.0082.10E0  ;3090  Z?,SC_0-K[.18]
U 10E0, 0000.003C.0180.0800.0000.126D  ;3091 =0 J/FOUNDIT
U 10E1, 0000.003C.0180.0800.0000.10DF  ;3092  J/KEEPLOOKING
            ;3093 FOUNDIT:
U 126D, 0000.003C.05F8.0800.0084.B26E  ;3094  SC_SC-K[.1],Q_0    ; SETUP TO SHIFT TO GET ID
U 126E, 0D00.003C.0180.0800.0000.126F  ;3095  D_DAL.SC
            ;3096  D_D.AND.K[.1F],    ; MASK AND SAVE CPU ID
U 126F, 0819.0036.8D80.0800.0000.0001  ;3097    RETURN1    ; Return to caller
```

```
;3098
```

```
;3099  .PAGE
;3100  .TOC "   CHECK INTERRUPT VECTOR ROUTINE"
;3101 ;*************************************************************
;3102 ;**
;3103 ;
;3104 ; Functional Description:
;3105 ; ---------------------
;3106 ;
;3107 ;      When certain SBI errors are being forced it may  be  desired
;3108 ; to check whether the correct interrupt vector was generated in ID
;3109 ; Register 0D (VECTOR).   When this subroutine is called,  Register
;3110 ; D should be holding the expected value of the VECTOR Register.
;3111 ;
;3112 ; Calling Constraint: =00
;3113 ; ------------------
;3114 ;
;3115 ;
;3116 ; Inputs:
;3117 ; ------
;3118 ;
;3119 ; D REG MUST HOLD THE EXPECTED VALUE OF THE VECTOR REGISTER
;3120 ;
;3121 ; Outputs:
;3122 ; -------
;3123 ;
;3124 ; IF THE CONTENTS OF THE VECTOR REGISTER ARE NOT
;3125 ; EQUAL TO THE EXPECTED, THE FOLLOWING DATA WILL
;3126 ; SAVED IN THE RC REGISTERS.
;3127 ;
;3128 ; RC[08] <- ID VECTOR
;3129 ; RC[07] <- SBI.ERR REGISTER
;3130 ; RC[06] <- SBI FAULT REGISTER
;3131 ;
;3132 ; Side Effects:
;3133 ; ------------
;3134 ;
;3135 ; CONTENTS OF THE FOLLOWING REGISTERS WILL BE LOST:
;3136 ;
;3137 ; D, Q
;3138 ;
;3139 ;--
;3140 ;*************************************************************
;3141 CHECK.VECTOR:
U 1270, 0000,003C,0180,0800,4000,1271  ;3142  IEK/ISTR    ; Strobe interrupt vector
U 1271, 0000,003C,35F0,2C00,0000,1272  ;3143  Q_ID[VECTOR]    ; Get contents of ID Vector Register
U 1272, 0019,2034,81C0,0800,0000,1273  ;3144  Q_Q.AND.K[.3FF]    ; Mask unwanted bits
        ;3145  ALU_D.XOR.Q,    ; See if the expected vector in D Reg
U 1273, 001D,0020,0180,0800,0010,1274  ;3146   CLK.UBCC   ; ...is equal to the received vector
        ;3147       ; ...in Q Reg
U 1274, 0000,013C,0180,0800,0000,10E2  ;3148  Z?     ; ...
U 10E2, 0000,003C,0180,0800,0000,1007  ;3149 =0 J/CHECK.INT.1    ; Vectors not equal. Go load the
        ;3150       ; ...error data
U 10E3, 0000,003E,0180,0800,0000,0001  ;3151  RETURN1     ; OK. Vectors are equal
```

```
            ;3152  .PAGE
            ;3153  .TOC "    ENABLE NEXT MS780-E"
            ;3154 ;++
            ;3155 ; FUNCTIONAL DESCRIPTION:
            ;3156 ;
            ;3157 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3158 ;    They will be tested  separately, according to the TR level at which
            ;3159 ;    they are located,  starting with the one  at the lowest level first.
            ;3160 ;          When  a test has finished  with the first unit, it will have to call
            ;3161 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3162 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3163 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3164 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3165 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3166 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3167 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3168 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3169 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3170 ;
            ;3171 ; CALLING CONSTRAINT:
            ;3172 ;
            ;3173 ;  =00
            ;3174 ;
            ;3175 ;--
            ;3176 ENABLE.NEXT.MS780E:
U 1275, 0000,003C,F9F0,2C00,0000,1276  ;3177  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3178  ALU Q,   ; Check to see if it is zero
U 1276, 0001,203C,0180,0800,0010,1277  ;3179     CLK.UBCC  ; Check Condition Codes
U 1277, 0000,013C,0180,0800,0000,10E4  ;3180  Z?   ; ...for zero
U 10E4, 0000,003C,0180,0800,0000,1278  ;3181 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10E5, 0000,003E,0180,0800,0000,0002  ;3182  RETURN2   ; Zero No more units to test
            ;3183 ENABLE.NEXT:
U 1278, 0818,0038,C180,0800,0000,10E6  ;3184  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10E6, 0000,003D,0180,0800,0000,1261  ;3185 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10E7, 0818,0038,7980,0800,0000,1279  ;3186  D_K[.30]  ; Prepare to point to the ID register holding
            ;3187     ; ...the I/O Base address of the next MS780-E
U 1279, 0819,0014,1180,0800,0000,127A  ;3188  D_D+K[.4]  ; ...
U 127A, 0000,003C,F580,0800,0000,127B  ;3189  K[.A]   ; ...
U 127B, 0819,0014,F580,0800,0000,127C  ;3190  D_D+K[.A]  ; ...
U 127C, 0000,003C,F9F0,2C00,0000,127D  ;3191  Q_ID[3E]  ; Get MS780-E Counter
            ;3192  D_D-Q,   ; D now holds the pointer to the ID register
U 127D, 081D,0000,0180,0800,0082,127E  ;3193   SC_ALU  ; ...
U 127E, 0000,003C,01F0,2400,0000,127F  ;3194  Q_ID(SC)  ; Q gets adrress of next MS780E
U 127F, 0001,203C,0180,09F0,0000,1280  ;3195  RC[0E]_Q  ; Save it also in RC[0E]
U 1280, 0F03,003C,0180,09E8,0000,10E8  ;3196  RC[0D]_0,D_0  ; Set starting address to zero
U 10E8, 0000,003D,0180,0800,0000,1261  ;3197 =0 CALL[SET.START.ADDR] ; ....
U 10E9, 0F00,003C,F9F0,2C00,0000,1281  ;3198  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1281, 081D,2008,0180,0800,0000,1282  ;3199  D_Q-D-1   ; Subtract 1 from total
U 1282, 0000,003C,F980,3C00,0000,10EA  ;3200  ID[3E]_D  ; Adjust the pointer
U 10EA, 0000,003D,0180,0800,0000,120C  ;3201 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10EB, 0000,003E,0180,0800,0000,0001  ;3202  RETURN1   ; Tell caller another unit was found
            ;3203
```

```
            ;3204   .PAGE
            ;3205   .TOC "    INIT SUBROUTINE"
            ;3206   ;=================================================================
            ;3207   ;
            ;3208   ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;3209   ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;3210   ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;3211   ;
            ;3212   ;=================================================================
            ;3213   ;
U 1283, 0003,003C,0180,0AF8,0000,10EC   ;3214 INIT: R[0F]_0     ; Clear u-traps
U 10EC, 0000,003D,0180,0800,0000,10C0   ;3215 =0 CALL[SBI.INIT]    ; Execute an unjam sequence
U 10ED, 0000,003C,0180,0800,0000,10EE   ;3216  NOP
U 10EE, 0000,003D,0180,0800,0000,1286   ;3217 =0 CALL[SETPSL]    ; Set psl and clear software
            ;3218       ; ...intrupt reg
U 10EF, 0000,003C,0180,0800,0000,10F0   ;3219  NOP
U 10F0, 0000,003D,0180,0800,0000,128D   ;3220 =0 CALL[CLRFLT]    ; Clear cpu fault bit
U 10F1, 0000,003C,0180,0800,0000,10F2   ;3221  NOP
U 10F2, 0000,003D,0180,0800,0000,1291   ;3222 =0 CALL[CLRECC]    ; Clear any pending ecc intrupts
U 10F3, 0000,003C,0180,0800,0000,10F4   ;3223  NOP
U 10F4, 0000,003D,0180,0800,0000,1289   ;3224 =0 CALL[CLRTIM]    ; Clear timeout bit
U 10F5, 0000,003C,0180,0800,0000,10F6   ;3225  NOP
U 10F6, 0000,003D,0180,0800,0000,1295   ;3226 =0 CALL[ENBECC]    ; Enable ecc intrupts
U 10F7, 0000,003C,0180,0800,0000,10F8   ;3227  NOP
U 10F8, 0000,003D,0180,0800,0000,1299   ;3228 =0 CALL[ENBFLT]    ; Enbale fault interrupts
U 10F9, 0818,0038,4580,0800,0000,10FA   ;3229  D_K[.8000]    ; Enable cache parity error
            ;3230 =0 CALL[DISABLE.CACHE],   ; Shut cache off
U 10FA, 0000,003D,7980,3C00,0000,1211   ;3231    ID[PARITY]_D
U 10FB, 0F00,003C,0180,0800,0000,1284   ;3232  D_0
U 1284, 0000,003C,4980,3C00,0000,1285   ;3233  ID[TBER0]_D    ; Shut TB off
U 1285, 0000,003E,7180,3C00,0000,0001   ;3234  ID[COMP]_D,RETURN1   ; Disable silo interrupts
```

```
            ;3235   .PAGE
            ;3236   .TOC "   SET PSL SUBROUTINE"
            ;3237   ;=============================================================================
            ;3238   ;
            ;3239   ;DROP THE CPU IPR LEVEL TO
            ;3240   ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;3241   ; PENDING
            ;3242   ;
            ;3243   ;=============================================================================
            ;3244   ;
U 1286. 0F00.003C.0180.0800.0000.1287   ;3245 SETPSL: D_0
U 1287. 0000.003C.3980.3C00.0000.1288   ;3246  ID[SIR]_D  ; Set software levels to zero
U 1288. 0000.003E.3D80.3C00.0000.0001   ;3247  RETURN1,ID[PSL]_D
```

```
            ;3248   .PAGE
            ;3249   .TOC "   CLEAR TIME OUT BIT SUBROUTINE"
            ;3250   ;=====================================================================
            ;3251   ;
            ;3252   ;CLEAR THE CP TIMEOUT BIT IN THE
            ;3253   ; SBI ERROR REGISTER
            ;3254   ;
            ;3255   ;=====================================================================
            ;3256   ;
U 1289, 0818,0038,0580,0800,0000,128A   ;3257 CLRTIM: D_K[.1]
U 128A, 0000,003C,85F8,0800,0084,728B   ;3258  Q_0,SC_K[.C]
U 128B, 0D00,003C,0180,0800,0000,128C   ;3259  D_DAL.SC
U 128C, 0000,003E,6580,3C00,0000,0001   ;3260  ID[SBI.ERR]_D,RETURN1
```

```
            ;3261   .PAGE
            ;3262   .TOC "    CLEAR FAULT REGISTER"
            ;3263   ;=================================================================
            ;3264   ;
            ;3265   ;CLEAR THE CP FAULT BIT IN THE
            ;3266   ; FAULT REGISTER
            ;3267   ;
            ;3268   ;=================================================================
            ;3269   ;
U 128D, 0818.0038.0180.0800.0000.128E   ;3270 CLRFLT: D_K[.8]
U 128E, 0000.003C.65F8.0800.0084.728F   ;3271   Q_0,SC_K[.10]
U 128F, 0D00.003C.0180.0800.0000.1290   ;3272   D_DAL.SC
U 1290, 0000.003E.6D80.3C00.0000.0001   ;3273   ID[FAULT]_D,RETURN1
```

```
            ;3274   .PAGE
            ;3275   .TOC "   CLEAR ECC IN SBI.ERR"
            ;3276   ;=================================================================
            ;3277   ;
            ;3278   ;CLEAR CRD.RDS BIT IN THE
            ;3279   ;SBI ERROR REGISTER
            ;3280   ;
            ;3281   ;=================================================================
            ;3282   ;
U 1291.  0818.0038.0D80.0800.0000.1292   ;3283 CLRECC: D_K[.3]
U 1292.  0000.003C.89F8.0800.0084.7293   ;3284  Q_0.SC_K[.D]
U 1293.  0D00.003C.0180.0800.0000.1294   ;3285  D_DAL.SC
U 1294.  0000.003E.6580.3C00.0000.0001   ;3286  ID[SBI.ERR]_D.RETURN1
```

```
            ;3287   .PAGE
            ;3288   .TOC "   ENABLE ECC, CRD/RDS INTRPTS"
            ;3289   ;=====================================================================
            ;3290   ;
            ;3291   ;ENABLE CRD,RDS INTERRUPTS
            ;3292   ;
            ;3293   ;=====================================================================
            ;3294   ;
U 1295. 0818.0038.0580.0800.0000.1296   ;3295 ENBECC: D_K[.1]
U 1296. 0000.003C.61F8.0800.0084.7297   ;3296  Q_0,SC_K[.F]
U 1297. 0D00.003C.0180.0800.0000.1298   ;3297  D_DAL.SC
U 1298. 0000.003E.6580.3C00.0000.0001   ;3298  ID[SBI.ERR]_D,RETURN1
            ;3299
            ;3300 ;*********************************************************************
            ;3301 ;*
            ;3302 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;3303 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;3304 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;3305 ;
            ;3306 ; D AND SC GET CLOBBERED
            ;3307 ;-
            ;3308 ;*********************************************************************
U 1299. 0F00.003C.2180.0800.0084.729A   ;3309 ENBFLT: SC_K[.14],D_0
U 129A. 0000.003C.0580.0800.0084.B29B   ;3310  SC_SC-K[.1]
U 129B. 0801.001C.0180.0800.0000.129C   ;3311  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 129C. 0600.003C.6D80.3C00.0000.129D   ;3312  ID[FAULT]_D,D_D.RIGHT
U 129D. 0000.003E.6D80.3C00.0000.0001   ;3313  ID[FAULT]_D,RETURN1
            ;3314
```

```
;3315  .PAGE "TEST 18D SBI FORCED FAULT TESTS"
;3316 ;**
;3317 ; FUNCTIONAL DESCRIPTION:
;3318 ;
;3319 ;          This test checks  the  fault  detection  logic  on  the  SBI
;3320 ;     interface board.  Each subtest will verify that the SBI Fault bit
;3321 ;     is cleared once the CPU deasserts the fault.  The fault  will  be
;3322 ;     checked in the CPU SBI Error/Fault Register as well as in the SBI
;3323 ;     Interface CNFG A Register.  This test will  not  check  for  data
;3324 ;     pattern sensitivity.
;3325 ;     The test starts out by generating constants.  For convenience and
;3326 ;     ease  of  programming  these will most likely be stored in cache.
;3327 ;     These constants will be used to mask undesired bits  in  the  SBI
;3328 ;     Error  and  Fault Registers or the CNFG Register A of the MS780-E
;3329 ;     under test.
;3330 ;
;3331 ;
;3332 ;     SUBTEST 1 - FAULT BITS CLEAR -
;3333 ;
;3334 ;          Check that the SBI Interface Fault and SBI Fault (in ID reg.
;3335 ;     1B,  SBI  Fault-Status)  bits  are initially clear.  If not, then
;3336 ;     attempt to clear them.
;3337 ;
;3338 ;
;3339 ;
;3340 ;     SUBTEST 2 - P0 FAULT -
;3341 ;
;3342 ;          Force P0 Fault using the SBI Maintenance register  (ID  reg.
;3343 ;     1D).   A  check is made to see that a SBI Fault Interrupt occured
;3344 ;     and that bit 31 of CNFG register A is set.
;3345 ;
;3346 ;
;3347 ;     SUBTEST 3 - P1 FAULT -
;3348 ;
;3349 ;          Force P1 Fault using the SBI  Maintenance  Register.   Check
;3350 ;     for  the  occurrence  of a SBI Fault Interrupt and that bit 31 of
;3351 ;     CNFG register A is set.
;3352 ;
;3353 ;
;3354 ;
;3355 ;     SUBTEST 4 - WRITE DATA SEQUENCE FAULT -
;3356 ;
;3357 ;          The SBI Maintenace register is used to force this fault  and
;3358 ;     bit 30 of CNFG register A is checked to see if it is set.
;3359 ;
;3360 ;
;3361 ;
;3362 ;     SUBTEST 5 - NORMAL INTERLOCK SEQUENCE -
;3363 ;
;3364 ;          The subtest verifies the correct operation of the  interlock
;3365 ;     sequence  logic,  by  executing  an Interlock Read followed by an
;3366 ;     Interlock Write, checking that no fault conditions are generated.
;3367 ;
;3368 ;
;3369 ;
```

```
;3370 ;    SUBTEST 6 - INTERLOCK SEQUENCE FAULT -
;3371 ;
;3372 ;         This subtest checks that an Interlock Write  that  does  not
;3373 ;    follow  an  Interlock  Read will cause a fault condition and that
;3374 ;    the logic will detect it.  A time-out will be expected due  to  a
;3375 ;    busy response from the memory.
;3376 ;
;3377 ;
;3378 ;    SUBTEST 7 - INTERLOCK TIMER TEST 1 -
;3379 ;
;3380 ;         Subtests 7 and 8 test the timing of the Interlock  Timer  on
;3381 ;    the  SBI Interface.  In subtest 7 two consecutive Interlock Reads
;3382 ;    are made.  The SBI  Interface  Interlock  Timer  is  expected  to
;3383 ;    time-out before the CPU does.
;3384 ;
;3385 ;
;3386 ;
;3387 ;    SUBTEST 8 - INTERLOCK TIMER TEST 2 -
;3388 ;
;3389 ;         In this subtest a normal Interlock Read  -  Interlock  Write
;3390 ;    sequence  is  executed  with a 100 usec (500 ticks) delay between
;3391 ;    the read and the write, checking that no Interlock Sequence Fault
;3392 ;    occurs.
;3393 ;
;3394 ; LOGIC DESCRIPTION:
;3395 ; -----------------
;3396 ;  N/A
;3397 ;
;3398 ; SYNC POINT DESCRIPTION:
;3399 ; ----------------------
;3400 ;  N/A
;3401 ;
;3402 ; ERROR DESCRIPTION:
;3403 ; -----------------
;3404 ; See individual error reports
;3405 ;
;3406 ;
;3407 ; REGISTER MAP:
;3408 ; ------------
;3409 ;
;3410 ; RA,B DESCRIPTION
;3411 ; ---- -----------
;3412 ;
;3413 ;    0 CLEAR CPU FAULT BIT MASK,    ^X0008 0000
;3414 ;   CPU FAULT BIT     ^X0008 0000
;3415 ;    1 FORCE P0 SBI ERROR      ^X8000 0000
;3416 ;   PARITY FAULT MASK    ^X8000 0000
;3417 ;    2 FORCE P1 SBI ERROR                    ^X0000 0800
;3418 ;    3 FORCE WRITE SEQ FAULT (INCL MAINT ID) ^X4F80 0000
;3419 ;    4 FORCE MULTIPLE XMITTERS (INCL MAINT ID) ^X1F80 0000
;3420 ;    5 MULTIPLE XMITTER FAULT MASK( REG A)  ^X0800 0000
;3421 ;    6 WRITE SEQUENCE FAULT BIT  ^X4000 0000
;3422 ;    7 CPU FAULT INTERRUPT ENABLE     ^X0004 0000
;3423 ;    8 SBI FAULT MASK     ^X0002 0000
;3424 ;    9 CLEAR FORCE ERROR BITS,DISABLE CACHE ^X0001 8000
```

```
            ;3425 ;   A TRANSMITTER DURING FAULT MASK  ^X0400 0000
            ;3426 ;   B MASK FOR CPU GOT BUS, DEVICE BUSY, TIMEOUT
            ;3427 ;      ^X0000 1C00
            ;3428 ;   C CONSTANT FOR CPU TIMEOUT, GOT BUS, DEVICE BUSY
            ;3429 ;      ^X0000 1400
            ;3430 ;   D DATA1  "0F0F0F0F"
            ;3431 ;   E SAVE UTRAP MASK
            ;3432 ;   F U-TRAPS FLAG
            ;3433 ;
            ;3434 ; RC DESCRIPTION
            ;3435 ; -- -----------
            ;3436 ;  2 DATA2  "F0F0F0F0
            ;3437 ;  3 INTERLOCK FAULT MASK (REG A)     ^X1000 0000
            ;3438 ;  9 FAULT BIT MASK    ^XFC00 0000
            ;3439 ;  E I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3440 ;  F NUMBER OF ARGUMENTS TO BE PASSED TO THE CONSOLE
            ;3441 ;
            ;3442 ;--
            ;3443 ;*****************************************************************************
            ;3444 =0
U 10FC. 0000.003D.0180.0800.0000.1063  ;3445 TEST.7: NEWTST   ;
U 10FD. 0000.003C.0180.0800.0000.102C  ;3446  NOP
            ;3447 =00
U 102C. 0000.003D.0180.0800.0000.10C8  ;3448  CALL[FIND.MS780E] ; Find all the MS780-E'S
            ;3449      ; ... on the SBI
U 102D. 0000.003C.0180.0800.0000.1396  ;3450  J/TEST.7.EXIT  ; NO MS780-E'S found
            ;3451 ;
            ;3452 ; Begin setting up the test constants
            ;3453 ;
U 102E. 0818.0038.F1F8.0800.0000.102F  ;3454  D_K[.FFFC],Q_0   ;RC 9 Fault Bit Mask
U 102F. 0000.003C.7D80.0800.0084.729E  ;3455  SC_K[.18]   ;
U 129E. 0D00.003C.0180.0800.0000.129F  ;3456  D_DAL.SC   ;
U 129F. 0001.003C.0180.09C8.0000.12A0  ;3457  RC[09]_D   ;
U 12A0. 0818.0038.45C0.0800.0000.12A1  ;3458  ALU_K[.8000],D_ALU,Q_ALU ;
U 12A1. 0000.003C.01A8.0800.0000.12A2  ;3459  Q_Q.LEFT   ;
U 12A2. 001D.0030.F5A8.0AC8.0084.72A3  ;3460  R[9]_D.OR.Q,SC_K[.A],Q_Q.LEFT ; R9  - Clear Force Error Bits
U 12A3. 0001.203C.05A8.0AC0.0084.92A4  ;3461  R[8]_Q,Q_Q.LEFT,SC_SC+K[.1] ; R8  - SBI Fault
U 12A4. 0D01.203C.01A8.0AB8.0000.12A5  ;3462  R[7]_Q,Q_Q.LEFT,D_DAL.SC ; R7  - Cpu Fault Interrupt Enable
U 12A5. 0001.203C.01A8.0A80.0000.12A6  ;3463  R[0]_Q,Q_Q.LEFT          ; R0  - Cpu Fault Bit
U 12A6. 0501.003C.0D80.0AD0.0084.72A7  ;3464  R[0A]_D,D_D.LEFT,SC_K[.3] ; R0A - Memory Transmitter during fault
U 12A7. 0D01.003C.0180.0AA8.0000.12A8  ;3465  R[5]_D,D_DAL.SC          ; R5  - Multiple Xmitter Fault
U 12A8. 0501.003C.0180.0AB0.0000.12A9  ;3466  R[6]_D,D_D.LEFT          ; R6  - Write Sequence Fault
U 12A9. 0C01.003C.0180.0A88.0000.12AB  ;3467  R[1]_D,D_Q    ; R1  - Force P0 SBI Error
U 12AB. 0B00.003C.0188.0800.0000.12AC  ;3468  D_D.SWAP,Q_Q.LEFT2  ;
U 12AC. 0041.003C.01A8.0A90.0000.12AD  ;3469  ALU_D,R[2]_ALU.RIGHT,Q_Q.LEFT ; R2  - Force P1 SBI Error
U 12AD. 0818.0038.8D80.0800.0000.12AE  ;3470  D_K[.1F]   ;
U 12AE. 0B00.003C.ED80.0800.0084.72AF  ;3471  D_D.SWAP,SC_K[.1B]  ;
U 12AF. 001D.0030.0580.0AA0.0084.92B0  ;3472  R[4]_D.OR.Q,SC_SC+K[.1]         ; R4  - Force Multiple Xmitters Fault
U 12B0. 0803.001C.0180.0A20.0000.12B1  ;3473  D_NOT.MASK,LAB_R[4]  ;
U 12B1. 0000.003C.0580.0800.0084.92B2  ;3474  SC_SC+K[.1]   ;
U 12B2. 0801.001C.0180.0800.0000.12B3  ;3475  D_D.ORNOT.MASK   ;
U 12B3. 001C.2014.0180.0A98.0084.72B4  ;3476  R[3]_LA+D,SC_K[.8]  ; R3  - Force Write Sequence Fault
U 12B4. 081B.0010.ED80.0800.0000.12B5  ;3477  ALU_0+K[.1B]+1,D_ALU  ;
U 12B5. 0D00.003C.0180.0800.0000.12B6  ;3478  D_DAL.SC   ;
U 12B6. 0001.003C.0180.0AD8.0000.12B7  ;3479  R[0B]_D    ; R0B - Cpu Time-Out Flag
```

```
U 12B7. 0818.0038.6580.0800.0000.12B8  ;3480  D_K[.10]   ;
U 12B8. 0B00.003C.0180.0800.0000.12B9  ;3481  D_D.SWAP   ;
U 12B9. 0001.003C.0180.0998.0000.12BA  ;3482  RC[3]_D    ; RC[3] - SBI Interlock Fault Mask
U 12BA. 0818.0038.CDF8.0800.0000.12BB  ;3483  D_K[.F0],Q_0   ; Prepare to set up RA- D, and E
U 12BB. 0000.003C.0180.0800.0084.72BC  ;3484  SC_K[.8]   ;
U 12BC. 0D00.003C.0180.0800.0000.12BD  ;3485  D_DAL.SC   ;
U 12BD. 0819.0030.CD80.0800.0000.12BE  ;3486  D_D.OR.K[.F0]   ;
U 12BE. 0D00.003C.0180.0800.0000.12BF  ;3487  D_DAL.SC   ;
U 12BF. 0819.0030.CD80.0800.0000.12C0  ;3488  D_D.OR.K[.F0]   ;
U 12C0. 0D00.003C.0180.0800.0000.12C1  ;3489  D_DAL.SC   ;
U 12C1. 0819.0030.CD80.0800.0000.12C2  ;3490  D_D.OR.K[.F0]   ;
U 12C2. 0001.003C.0180.0AE8.0000.12C3  ;3491  R[0D]_D    ; RA - D Set up DATA1
U 12C3. 0801.0028.0180.0800.0000.12C4  ;3492  D_NOT.D    ;
U 12C4. 0001.003C.0180.0990.0000.12C5  ;3493  RC[02]_D   ; RC[02] Set up DATA2
U 12C5. 0F00.003C.8580.0800.0084.72C6  ;3494  D_0.SC_K[.C]   ; Prepare to set up R[0C]
U 12C6. 0801.001C.0180.0800.0000.12C7  ;3495  D_D.ORNOT.MASK   ;
U 12C7. 0000.003C.0980.0800.0084.B2C8  ;3496  SC_SC-K[.2]   ;
U 12C8. 0801.001C.0180.0800.0000.12C9  ;3497  D_D.ORNOT.MASK   ;
U 12C9. 0001.003C.0180.0AE0.0000.1030  ;3498  R[0C]_D    ; Cpu Time-Out Busy Response
        ;3499 =00
        ;3500 RESTART.TEST.7:    ; Entry point for second controller
        ;3501       ; ... and or next MS780-E
U 1030. 0000.003D.0180.0800.0000.10B6  ;3502  CALL[START.TEST]  ; Configure Controllers
U 1031. 0000.003C.0180.0800.0000.1396  ;3503  J/TEST.7.EXIT   ; No more controllers or MS780-E'S
U 1032. 0000.003C.0180.0800.0000.10FE  ;3504  nop    ;
        ;3505 =
        ;3506 ;////////////////////////////////////////////////////////////////////////////
        ;3507 ;
        ;3508 ;  SUBTEST 1
        ;3509 ;
U 10FE. 0000.003D.0180.0800.0000.11F0  ;3510 =0 SUBTEST    ; Start Subtest 1
U 10FF. 0000.003C.0180.0800.0000.110A  ;3511  nop    ;
U 110A. 0000.003D.0180.0800.0000.11E4  ;3512 =0 ERLOOP    ; Pass Uaddress for loop on error
U 110B. 0000.003C.0180.0800.0000.1110  ;3513  nop    ;
U 1110. 0000.003D.0180.0800.0000.1283  ;3514 =0 CALL[INIT]   ; Clear the SBI
U 1111. 0000.003C.6DF0.2C00.0000.12CA  ;3515  Q_ID[FAULT]   ; Check for SBI Fault Clear
U 12CA. 000D.2034.0180.0A40.0010.12CB  ;3516  ALU_Q[AND]R[8],CLK.UBCC          ; Test SBI Fault Bit
U 12CB. 0000.013C.0180.0800.0000.1112  ;3517  Z?    ; Check condition codes
```

```
            ;3518  .PAGE
            ;3519 ;////////////////////////////////////////////////////////////////////////////////
            ;3520 ;
U 1112. 0000.003D.1980.0800.0084.70B0  ;3521 =0 ERROR1[ZERO]   ; SBI Fault still asserted
            ;3522 ;
            ;3523 ;////////////////////////////////////////////////////////////////////////////////
            ;3524 ;
            ;3525 ; ERROR LOGOUT:
            ;3526 ;
            ;3527 ; DATA:  NO DATA LOGOUT
            ;3528 ;
            ;3529 ;////////////////////////////////////////////////////////////////////////////////
            ;3530 ;
U 1113. 000D.2034.0180.0A00.0010.12CC  ;3531  ALU_Q[AND]R[0],CLK.UBCC  ; Cpu Fault Clear ?
U 12CC. 0000.013C.0180.0800.0000.1114  ;3532  Z?    ; Check condition codes
```

```
            ;3533  .PAGE
            ;3534 ;
            ;3535 ;////////////////////////////////////////////////////////////////////////////////
            ;3536 ;
U 1114, 0000,003D,1980,0800,0084,70B0  ;3537 =0 ERROR1[ZERO]   ;Cpu Fault Bit Set
            ;3538 ;
            ;3539 ;////////////////////////////////////////////////////////////////////////////////
            ;3540 ;
            ;3541 ; ERROR LOGOUT:
            ;3542 ;
            ;3543 ; DATA: NO DATA LOGOUT
            ;3544 ;
            ;3545 ;////////////////////////////////////////////////////////////////////////////////
            ;3546 ;
U 1115, 0011,2034,01C0,0948,0000,12CD  ;3547  Q_ALU,ALU_Q[AND]RC[09]   ; Check all SBI Fault Status Bits
            ;3548          ; ... for clear
U 12CD, 0001,203C,0180,09E0,0010,12CE  ;3549  RC[0C]_Q,CLK.UBCC  ; Bits in error to console
U 12CE, 0003,013C,0180,09E8,0000,1116  ;3550  Z?,RC[0D]_0
```

```
            ;3551   .PAGE
            ;3552 ;
            ;3553 ;////////////////////////////////////////////////////////////////////////////////
            ;3554 ;
U 1116. 0000.003D.0D80.0800.0084.70B0  ;3555 =0 ERROR1[.3]    ; Some bits set
            ;3556
            ;3557 ;
            ;3558 ;////////////////////////////////////////////////////////////////////////////////
            ;3559 ;
            ;3560 ; ERROR LOGOUT:
            ;3561 ;
            ;3562 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3563 ;  EXPECTED SBI FAULT BITS  (31-26)
            ;3564 ;  RECIEVED SBI FAULT BITS  (31-26)
            ;3565 ;
            ;3566 ;////////////////////////////////////////////////////////////////////////////////
            ;3567 ;
U 1117. 0010.0038.0180.0970.0200.12CF  ;3568  VA_RC[0E]    ; Check Memory Configuration REG A
            ;3569       ; ...for SBI Fault Status Bits clear
U 12CF. 0000.003C.0180.C800.0000.12D0  ;3570  D[LONG]_CACHE.P    ; Read contents of CNFG A
U 12D0. 0001.003C.0180.09E8.0000.1034  ;3571  RC[0D]_D    ; Save the data read
U 1034. 0000.003D.0180.0800.0000.121A  ;3572 =00 CHECK.INTERRUPT    ; Check for interrupts
U 1035. 0000.003C.0180.0800.0000.12D1  ;3573  J/TEST7.S1.NOINT  ; Jump if no interrupt else
```

```
            ;3574  .PAGE
            ;3575 ;
            ;3576 ;////////////////////////////////////////////////////////////////////////////////
            ;3577 ;
U 1036. 0000.003D.F580.0800.0084.70B0  ;3578        ERROR1[.A]    ; UNEXPECTED INTERRUPT !!!!!
            ;3579 ;
            ;3580 ;
            ;3581 ;////////////////////////////////////////////////////////////////////////////////
            ;3582 ;
            ;3583 ; ERROR LOGOUT:
            ;3584 ;
            ;3585 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3586 ;  CONTENTS OF CNFG REG A
            ;3587 ;  NOT USED
            ;3588 ;  NOT USED
            ;3589 ;  NOT USED
            ;3590 ;  NOT USED
            ;3591 ;  INTERRUPT VECTOR REGISTER
            ;3592 ;  SBI.ERROR REGISTER
            ;3593 ;  SBI.FAULT REGISTER
            ;3594 ;  INTERRUPT OCCURED FLAG (TRUE=1,FALSE=0)
            ;3595 ;////////////////////////////////////////////////////////////////////////////////
            ;3596 =
            ;3597 TEST7.S1.NOINT:
U 12D1. 0010.0038.01C0.0968.0000.12D2  ;3598  Q_RC[0D]    ; Get data read
U 12D2. 0011.2034.01C0.0948.0010.12D3  ;3599  Q_ALU,ALU_Q[AND]rc[09],CLK.UBCC ; Check fault bits for clear
U 12D3. 0001.203C.0180.0AE0.0010.12D4  ;3600  R[0C]_Q,CLK.UBCC  ;
U 12D4. 0003.013C.0180.09E8.0000.1118  ;3601  Z?,RC[0D]_0    ; Check condition codes
```

```
          ;3602  .PAGE
          ;3603 ;
          ;3604 ;////////////////////////////////////////////////////////////////////////////////
          ;3605 ;
U 1118. 0000.003D.0D80.0800.0084.70B0  ;3606 =0 ERROR1[.3]   ;  SBI Fault bits set in  CNFG Reg A
          ;3607 ;
          ;3608 ;////////////////////////////////////////////////////////////////////////////////
          ;3609 ;
          ;3610 ; ERROR LOGOUT:
          ;3611 ;
          ;3612 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
          ;3613 ;  EXPECTED SBI FAULT BITS  in CNFG REG A
          ;3614 ;  RECIEVED SBI FAULT BITS  IN CNFG REG A
          ;3615 ;
          ;3616 ;////////////////////////////////////////////////////////////////////////////////
U 1119. 0000.003C.0180.0800.0000.111A  ;3617  NOP
          ;3618 ;////////////////////////////////////////////////////////////////////////////////
          ;3619 ;
          ;3620 ;   SUBTEST 2  FORCE PO FAULT
          ;3621 ;
          ;3622 ;
          ;3623 ; SEQUENCE OF STEPS
          ;3624 ; 1)CACHE OFF,FORCE PO FAULT
          ;3625 ; 2)INITIATE WRITE OPERATION
          ;3626 ; 3)CLEAR FORCE PO FAULT
          ;3627 ; 4)SEE SBI FAULT INTERRUPT
          ;3628 ; 5)SEE CPU FAULT SET
          ;3629 ; 6)SEE SBI FAULT SET
          ;3630 ; 7)SEE PARITY BIT SET IN CPU FAULT REG
          ;3631 ;   AND IN MEMORY CONFIGURATION A REGISTER
          ;3632 ; 8)DEASSERT FAULT BIT AND CHECK FOR FAULT BITS TO CLEAR
          ;3633 ;
          ;3634 ;********************************************************************************
          ;3635 ;
U 111A. 0000.003D.0180.0800.0000.11F0  ;3636 =0 SUBTEST    ; Increment subtest number
U 111B. 0000.003C.0180.0800.0000.111C  ;3637  NOP    ;
U 111C. 0000.003D.0180.0800.0000.11E4  ;3638 =0 ERLOOP    ; Set the error loop
U 111D. 0000.003C.0180.0800.0000.111E  ;3639  NOP    ;
U 111E. 0000.003D.0180.0800.0000.1283  ;3640 =0 CALL[INIT]   ; Initialize the SBI
U 111F. 0800.003C.0180.0A48.0000.12D5  ;3641  D_R[9]    ; Force the cache off,force P0 Fault
U 12D5. 080D.0030.0180.0A08.0000.12D6  ;3642  D_ALU,ALU_D[OR]R[1]  ;
U 12D6. 0000.003C.7580.3C00.0000.12D7  ;3643  ID[MAINT]_D   ; Load SBI Maintenance Register
U 12D7. 0F00.003C.0180.0800.0000.12D8  ;3644  D_0    ; Get data
U 12D8. 0001.003C.0180.0800.0200.12D9  ;3645  VA_D    ; Point to location 0
U 12D9. 0000.003C.0180.A800.0000.12DA  ;3646  CACHE.P_D[LONG]   ; Force the error
U 12DA. 0000.003C.0180.0800.0000.12DB  ;3647  NOP    ;
U 12DB. 0000.003C.0180.0800.0000.12DC  ;3648  NOP    ;
U 12DC. 0000.003C.0180.0800.0000.12DD  ;3649  NOP    ;CLOCK TICS
U 12DD. 0000.003C.0180.0800.0000.12DE  ;3650  NOP    ;
U 12DE. 0000.003C.0180.0800.0000.12DF  ;3651  NOP    ;
U 12DF. 0000.003C.0180.0800.0000.12E0  ;3652  NOP    ;
U 12E0. 0800.003C.0180.0A48.0000.12E1  ;3653  D_R[9]    ; Clear the Force Fault Bit
U 12E1. 0000.003C.7580.3C00.0000.1120  ;3654  ID[MAINT]_D   ; ...IN ID Maintenance register
          ;3655 =0 D_K[.FF],   ; Set up number of cycles to wait
U 1120. 0818.0039.4980.0B00.0000.1219  ;3656   CALL[WAIT.LOOP]  ; Wait for SBI to settle
```

```
U 1121, 0818,0038,3580,0800,0000,12E2  ;3657  D_K[.50]   ; Generate the expected interrupt
U 12E2, 0819,0030,8580,0800,0000,12E3  ;3658  D_D.OR.K[.C]   ; ....vector
U 12E3, 0001,003C,0180,09E8,0000,1038  ;3659  RC[0D]_D   ; Load expected Interrupt vector
U 1038, 0000,003D,0180,0800,0000,1270  ;3660 =00 CALL[CHECK.VECTOR]  ; Was the correct Interrupt Vector
      ;3661      ; ...generated ?
U 1039, 0000,003C,0180,0800,0000,12E4  ;3662  J/VECTOK.1   ; Ok. Vector was generated correctly
```

```
            ;3663  .PAGE
            ;3664 ;
            ;3665 ;////////////////////////////////////////////////////////////////////////////////
            ;3666 ;
U 103A, 0000,003D,D980,0800,0084,70B0  ;3667  ERROR1[.9]   ; Wrong vector was generated
            ;3668
            ;3669 ;
            ;3670 ;////////////////////////////////////////////////////////////////////////////////
            ;3671 ;
            ;3672 ; ERROR LOGOUT:
            ;3673 ;
            ;3674 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3675 ;  EXPECTED INTERRUPT VECTOR
            ;3676 ;  NOT USED
            ;3677 ;  NOT USED
            ;3678 ;  NOT USED
            ;3679 ;  NOT USED
            ;3680 ;  ID VECTOR REGISTER
            ;3681 ;  ID SBI ERROR REGISTER
            ;3682 ;  ID FAULT REGISTER
            ;3683 ;
            ;3684 ;
            ;3685 ;////////////////////////////////////////////////////////////////////////////////
            ;3686 ;
            ;3687 =
            ;3688 VECTOK.1:
U 12E4, 0000,003C,6DF0,2C00,0000,12E5  ;3689  Q_ID[FAULT]   ; Check CPU Fault Bit Set
U 12E5, 0001,203C,0180,09E8,0000,12E6  ;3690  RC[0D]_Q   ; Save for error logout
U 12E6, 000D,2034,0180,0A00,0010,12E7  ;3691  ALU_Q[AND]R[0],CLK.UBCC
U 12E7, 0000,013C,0180,0800,0000,1122  ;3692  Z?
U 1122, 0000,003C,0180,0800,0000,12E8  ;3693 =0 J/T7.CFBIT.OK   ; O.K. BIT SET
```

```
            ;3694  .PAGE
            ;3695 ;
            ;3696 ;////////////////////////////////////////////////////////////////////////////////
            ;3697 ;
U 1123. 0000.003D.0980.0800.0084.70B0  ;3698   ERROR1[.2]    ; CPU Fault is clear
            ;3699 ;       ; ...should be set
            ;3700 ;
            ;3701 ;////////////////////////////////////////////////////////////////////////////////
            ;3702 ;
            ;3703 ; ERROR LOGOUT:
            ;3704 ;
            ;3705 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3706 ;  ID FAULT REGISTER
            ;3707 ;
            ;3708 ;////////////////////////////////////////////////////////////////////////////////
            ;3709 ;
            ;3710
            ;3711 T7.CFBIT.OK:
U 12E8. 000D.2034.0180.0A40.0010.12E9  ;3712  ALU_Q[AND]R[8],CLK.UBCC          ; Check SBI Fault Bit Set
U 12E9. 0000.013C.0180.0800.0000.1124  ;3713  Z?
U 1124. 0000.003C.0180.0800.0000.12EA  ;3714 =0 J/T7.SFBIT.OK    ; OK .. SBI FAULT SET
```

```
            ;3715  .PAGE
            ;3716 ;
            ;3717 ;////////////////////////////////////////////////////////////////////////////////
            ;3718 ;
U 1125, 0000.003D.0980.0800.0084.70B0  ;3719     ERROR1[.2]    ; SBI Parity Fault CLEAR shoud
            ;3720 ;       ; ...  be SET
            ;3721 ;
            ;3722 ;
            ;3723 ;////////////////////////////////////////////////////////////////////////////////
            ;3724 ;
            ;3725 ; ERROR LOGOUT:
            ;3726 ;
            ;3727 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3728 ;  ID FAULT REGISTER
            ;3729 ;
            ;3730 ;////////////////////////////////////////////////////////////////////////////////
            ;3731 ;
            ;3732
            ;3733 T7.SFBIT.OK:
U 12EA, 0011.2034.01C0.0948.0010.12EB  ;3734  Q_ALU,ALU_Q[AND]rc[09],clk.ubcc ; Did CPU see parity fault
U 12EB, 0001.203C.0180.09E8.0000.12EC  ;3735  RC[0D]_Q    ; Save for error logout
U 12EC, 0800.003C.0180.0A08.0000.12ED  ;3736  D_R[1]     ; Set Parity Bit
U 12ED, 080D.0030.0180.0A50.0000.12EE  ;3737  D_ALU,ALU_D[OR]R[0A]   ; Was CPU the Transmitter
U 12EE, 001D.2020.0180.0800.0010.12EF  ;3738  ALU_Q[XOR]D,CLK.UBCC      ; Did CPU see Parity Error and
U 12EF, 0001.013C.0180.09E0.0000.1126  ;3739  Z?,RC[0C]_d    ; ...    as the Transmitter
```

```
            ;3740   .PAGE
            ;3741 ;
            ;3742 ;////////////////////////////////////////////////////////////////////////////////
            ;3743 ;
U 1126. 0000.003D.0D80.0800.0084.70B0  ;3744 =0 ERROR1[.3]     ; NO wrong bits set
            ;3745 ;       ; Bit 26 in ID fault Not set
            ;3746 ;
            ;3747 ;////////////////////////////////////////////////////////////////////////////////
            ;3748 ;
            ;3749 ; ERROR LOGOUT:
            ;3750 ;
            ;3751 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3752 ;  EXPECTED SBI-FAULT-STATUS BITS (31-26)
            ;3753 ;  MASKED SBI-FAULT-STATUS BITS (31-26)
            ;3754 ;
            ;3755 ;////////////////////////////////////////////////////////////////////////////////
            ;3756 ;
            ;3757
U 1127. 0010.0038.0180.0970.0200.12F0  ;3758  VA_RC[0E]    ; Did memory see fault ?
U 12F0. 0000.003C.0180.C800.0000.12F1  ;3759  D[LONG]_CACHE.P    ; Read Config A Reg
U 12F1. 0811.0034.0180.0948.0000.12F2  ;3760  D_ALU,ALU_D[AND]RC[09]   ; Mask fault bits
U 12F2. 0001.003C.0180.09D8.0000.12F3  ;3761  RC[0B]_D    ; Save masked image
U 12F3. 0818.0038.3580.0800.0000.12F4  ;3762  D_K[.50]    ; Generate the expected interrupt
U 12F4. 0819.0030.8580.0800.0000.12F5  ;3763  D_D.OR.K[.C]    ; ....vector
U 12F5. 0001.003C.0180.09E8.0000.103C  ;3764  RC[0D]_D    ; Save expected vector for error logout
U 103C. 0000.003D.0180.0800.0000.1270  ;3765 =00 CALL[CHECK.VECTOR]   ; Was the correct Interrupt Vector
            ;3766       ; ...generated ?
U 103D. 0000.003C.0180.0800.0000.12F6  ;3767  J/VECTOK.2    ; Ok. Correct vector generated
```

```
            ;3768  .PAGE
            ;3769 ;
            ;3770 ;////////////////////////////////////////////////////////////////////////////////
            ;3771 ;
U 103E, 0000,003D,D980,0800,0084,70B0  ;3772  ERROR1[.9]   ; Wrong vector was generated
            ;3773
            ;3774 ;
            ;3775 ;////////////////////////////////////////////////////////////////////////////////
            ;3776 ;
            ;3777 ; ERROR LOGOUT:
            ;3778 ;
            ;3779 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3780 ;  EXPECTED INTERRUPT VECTOR
            ;3781 ;  NOT USED
            ;3782 ;  NOT USED
            ;3783 ;  NOT USED
            ;3784 ;  NOT USED
            ;3785 ;  ID VECTOR REGISTER
            ;3786 ;  ID SBI ERROR REGISTER
            ;3787 ;  ID FAULT REGISTER
            ;3788 ;
            ;3789 ;
            ;3790 ;////////////////////////////////////////////////////////////////////////////////
            ;3791 ;
            ;3792 =
            ;3793 VECTOK.2:
U 12F6, 0000,003C,01C0,0A08,0000,12F7  ;3794  Q_R[1]     ; Get parity fault mask
U 12F7, 0011,2020,0180,0958,0010,12F8  ;3795  ALU_Q[XOR]RC[0B],CLK.UBCC ; Check the Parity Fault Bit
U 12F8, 0001,213C,0180,09E0,0000,1128  ;3796  Z?,RC[0C]_Q
```

```
            ;3797  .PAGE
            ;3798 ;
            ;3799 ;////////////////////////////////////////////////////////////////////////////////
            ;3800 ;
U 1128. 0000.003D.1180.0800.0084.70B0  ;3801 =0 ERROR1[.4]   ; Wrong Fault bits set
            ;3802 ;
            ;3803 ;
            ;3804 ;////////////////////////////////////////////////////////////////////////////////
            ;3805 ;
            ;3806 ; ERROR LOGOUT:
            ;3807 ;
            ;3808 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3809 ;  NOT USED
            ;3810 ;  EXPECTED CONTENTS OF CNFG A REGISTER
            ;3811 ;  RECIEVED CONTENTS OF CNFG A REGISTER
            ;3812 ;
            ;3813 ;////////////////////////////////////////////////////////////////////////////////
            ;3814 ;
U 1129. 0000.003C.0180.0800.0000.112A  ;3815  NOP
U 112A. 0000.003D.0180.0800.0000.10C0  ;3816 =0 CALL[SBI.INIT]   ; Clear the sbi fault bits
U 112B. 0000.003C.6DF0.2C00.0000.12F9  ;3817  Q_ID[FAULT]   ; ID Fault Register to Q
U 12F9. 0001.203C.0180.09E8.0000.12FA  ;3818  RC[0D]_Q   ; Save in RC[0D] for error logout
U 12FA. 000D.2034.0180.0A40.0010.12FB  ;3819  ALU_Q[AND]R[8],CLK.UBCC  ; SBI Fault Clear  ?
U 12FB. 0000.013C.0180.0800.0000.112C  ;3820  Z?    ; Check condition codes
```

```
            ;3821  .PAGE
            ;3822 ;
            ;3823 ;////////////////////////////////////////////////////////////////////////////////
            ;3824 ;
U 112C. 0000.003D.0980.0800.0084.70B0  ;3825 =0 ERROR1[.2]    ; NO SBI Fault Bit is set
            ;3826 ;
            ;3827 ;////////////////////////////////////////////////////////////////////////////////
            ;3828 ;
            ;3829 ; ERROR LOGOUT:
            ;3830 ;
            ;3831 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3832 ;  ID FAULT REGISTER
            ;3833 ;
            ;3834 ;////////////////////////////////////////////////////////////////////////////////
            ;3835 ;
U 112D. 000D.2034.0180.0A00.0010.12FC  ;3836  ALU_Q[AND]R[0],CLK.UBCC          ; CPU Fault Clear ?
U 12FC. 0000.013C.0180.0800.0000.112E  ;3837  Z?
```

```
            ;3838  .PAGE
            ;3839 ;
            ;3840 ;////////////////////////////////////////////////////////////////////////////////
            ;3841 ;
U 112E. 0000.003D.0980.0800.0084.70B0  ;3842 =0 ERROR1[.2]    ; NO CPU Fault did NOT clear
            ;3843 ;
            ;3844 ;
            ;3845 ;////////////////////////////////////////////////////////////////////////////////
            ;3846 ;
            ;3847 ; ERROR LOGOUT:
            ;3848 ;
            ;3849 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3850 ;  ID FAULT REGISTER
            ;3851 ;
            ;3852 ;////////////////////////////////////////////////////////////////////////////////
            ;3853 ;
U 112F. 0011.2034.01C0.0948.0000.12FD  ;3854  Q_ALU,ALU_Q[AND]rc[09]  ;CPU SBI FAULT STATUS BITS CLEAR ?
U 12FD. 0001.203C.0180.09E8.0010.12FE  ;3855  RC[0D]_Q,CLK.UBCC
U 12FE. 0003.013C.0180.09E0.0000.1130  ;3856          Z?,RC[0C]_0
```

```
            ;3857  .PAGE
            ;3858 ;
            ;3859 ;////////////////////////////////////////////////////////////////////////////
            ;3860 ;
U 1130, 0000,003D,0D80,0800,0084,70B0  ;3861 =0 ERROR1[.3]   ;CPU SBI-STATUS DID NOT CLEAR
            ;3862 ;
            ;3863 ;////////////////////////////////////////////////////////////////////////////
            ;3864 ;
            ;3865 ; ERROR LOGOUT:
            ;3866 ;
            ;3867 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3868 ;  MASKED SBI-FAULT STATUS BITS READ BITS (31-26)
            ;3869 ;  EXPECTED SBI_FAULT STATUS BITS (31-26)
            ;3870 ;
            ;3871 ;////////////////////////////////////////////////////////////////////////////
            ;3872 ;
U 1131, 0010,0038,0180,0970,0200,12FF  ;3873  VA_RC[0E]
U 12FF, 0000,003C,0180,C800,0000,1300  ;3874  D[LONG]_CACHE.P   ; Read CNFG A save in  D Register
U 1300, 0811,0034,0180,0948,0000,1301  ;3875  D_ALU,ALU_D[AND]rc[09]   ; Mask Fault Bits
U 1301, 0001,003C,0180,09E8,0000,1302  ;3876  RC[0D]_D   ; Save Masked Bits
U 1302, 0810,0038,0180,0968,0010,1303  ;3877  D_RC[0D],CLK.UBCC
U 1303, 0000,013C,0180,0800,0000,1132  ;3878  Z?
```

```
            ;3879  .PAGE
            ;3880 ;
            ;3881 ;////////////////////////////////////////////////////////////////////////////////
            ;3882 ;
U 1132, 0000,003D,0D80,0800,0084,70B0  ;3883 =0 ERROR1[.3]     ; SBI Fault Status Bits set
            ;3884 ;       ; ... should be cleared
            ;3885 ;
            ;3886 ;////////////////////////////////////////////////////////////////////////////////
            ;3887 ;
            ;3888 ; ERROR LOGOUT:
            ;3889 ;
            ;3890 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3891 ;  MASKED FAULTS BITS FROM CNFG A REGISTER
            ;3892 ;  EXPECTED FAULTS BITS
            ;3893 ;
            ;3894 ;////////////////////////////////////////////////////////////////////////////////
            ;3895 ;
            ;3896 ;** SUBTEST 3 **
            ;3897 ;
            ;3898 ; CHECK P1 FAULT LOGIC
            ;3899 ;
            ;3900 ; SEQUENCE OF STEPS
            ;3901 ; 1) LOAD MAINTENANCE REGISTER WITH FORCE P1 ERROR
            ;3902 ; 2) WRITE TO MEMORY
            ;3903 ; 3) CHECK FOR SBI PARITY FAULT IN CPU
            ;3904 ; 4) CLEAR FORCE P1 IN MAINTENANCE
            ;3905 ; 5) READ MEMORY CONFIG REG A SEE PARITY FAULT
            ;3906 ; 7) SEE MEMORY CONFIG REG A PARITY FAULT CLEARED
            ;3907 ;
            ;3908 ;////////////////////////////////////////////////////////////////////////////////
            ;3909 ;
U 1133, 0000,003C,0180,0800,0000,1134  ;3910  NOP
U 1134, 0000,003D,0180,0800,0000,11F0  ;3911 =0 SUBTEST     ; Increment subtest number
U 1135, 0000,003C,0180,0800,0000,1136  ;3912  NOP      ; ...
U 1136, 0000,003D,0180,0800,0000,11E4  ;3913 =0 ERLOOP     ; Set up Uaddress for error loop
U 1137, 0000,003C,0180,0800,0000,1138  ;3914  NOP      ; ...
U 1138, 0000,003D,0180,0800,0000,1283  ;3915 =0 CALL[INIT]    ; Initialize the SBI
U 1139, 0800,003C,0180,0A10,0000,1304  ;3916  D_R[2]     ; Generate cache off and force
U 1304, 080D,0030,0180,0A48,0000,1305  ;3917  D_ALU,ALU_D[OR]R[9]  ; ... P1 fault
U 1305, 0000,003C,7580,3C00,0000,1306  ;3918  ID[MAINT]_D    ; Set bits in ID MAINT Reg
U 1306, 0F00,003C,0180,0800,0000,1307  ;3919  D_0     ; Point to address 0
U 1307, 0001,003C,0180,0800,0200,1308  ;3920  VA_D     ; ... in cache
U 1308, 0000,003C,0180,A800,0000,1309  ;3921  CACHE.P_D[LONG]    ; Do the write operation
U 1309, 0000,003C,0180,0800,0000,130A  ;3922  NOP     ; Wait for the SBI to settle
U 130A, 0000,003C,0180,0800,0000,130B  ;3923  NOP     ; ...
U 130B, 0000,003C,0180,0800,0000,130C  ;3924  NOP     ; ...
U 130C, 0000,003C,0180,0800,0000,130D  ;3925  NOP     ; CLOCK TICS
U 130D, 0000,003C,0180,0800,0000,130E  ;3926  NOP     ; ...
U 130E, 0000,003C,0180,0800,0000,130F  ;3927  NOP     ; ...
U 130F, 0000,003C,0180,0800,0000,1310  ;3928  NOP     ; ...
U 1310, 0800,003C,0180,0A48,0000,1311  ;3929  D_R[9]     ; Clear Force Fault Bit
U 1311, 0000,003C,7580,3C00,0000,113A  ;3930  ID[MAINT]_D    ; Set bits in ID Maint Reg
            ;3931 =0 D_K[.FF],    ; Set up number of cycles to wait
U 113A, 0818,0039,4980,0800,0000,1219  ;3932   CALL[WAIT.LOOP]  ; Wait for SBI to settle
U 113B, 0818,0038,3580,0800,0000,1312  ;3933  D_K[.50]    ; Generate the expected interrupt
```

```
U 1312, 0819,0030,8580,0800,0000,1313  ;3934  D_D.OR.K[.C]   ; ....vector
U 1313, 0001,003C,0180,09E8,0000,1040  ;3935  RC[0D]_D   ; Save expected vector for error logout
U 1040, 0000,003D,0180,0800,0000,1270  ;3936 =00 CALL[CHECK.VECTOR]  ; Was the correct Interrupt Vector
        ;3937      ; ...generated ?
U 1041, 0000,003C,0180,0800,0000,1314  ;3938  J/VECTOK.3   ; OK. Correct vector was generated
```

```
            ;3939  .PAGE
            ;3940 ;
            ;3941 ;////////////////////////////////////////////////////////////////////////////////
            ;3942 ;
U 1042, 0000,003D,D980,0800,0084,70B0  ;3943  ERROR1[.9]   ; Wrong vector was generated
            ;3944
            ;3945 ;
            ;3946 ;////////////////////////////////////////////////////////////////////////////////
            ;3947 ;
            ;3948 ; ERROR LOGOUT:
            ;3949 ;
            ;3950 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3951 ;  EXPECTED INTERRUPT VECTOR
            ;3952 ;  NOT USED
            ;3953 ;  NOT USED
            ;3954 ;  NOT USED
            ;3955 ;  NOT USED
            ;3956 ;  ID VECTOR REGISTER
            ;3957 ;  ID SBI ERROR REGISTER
            ;3958 ;  ID FAULT REGISTER
            ;3959 ;
            ;3960 ;
            ;3961 ;////////////////////////////////////////////////////////////////////////////////
            ;3962 ;
            ;3963 =
            ;3964 VECTOK.3:
U 1314, 0000,003C,6DF0,2C00,0000,1315  ;3965  Q_ID[FAULT]   ; Check CPU Fault Bit Set
U 1315, 0001,203C,0180,09E8,0000,1316  ;3966  RC[0D]_Q   ; Save for error logout
U 1316, 000D,2034,0180,0A00,0010,1317  ;3967  ALU_Q[AND]R[0],CLK.UBCC
U 1317, 0000,013C,0180,0800,0000,113C  ;3968  Z?
U 113C, 0000,003C,0180,0800,0000,1318  ;3969 =0 J/T7.CFBIT.OK3   ; O.K. BIT SET
```

```
            ;3970  .PAGE
            ;3971 ;
            ;3972 ;////////////////////////////////////////////////////////////////////////////////
            ;3973 ;
U 113D. 0000.003D.0980.0800.0084.70B0  ;3974   ERROR1[.2]    ; CPU Fault was clear
            ;3975 ;      ; ...should be set
            ;3976 ;
            ;3977 ;////////////////////////////////////////////////////////////////////////////////
            ;3978 ;
            ;3979 ; ERROR LOGOUT:
            ;3980 ;
            ;3981 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3982 ;  ID FAULT REGISTER
            ;3983 ;
            ;3984 ;////////////////////////////////////////////////////////////////////////////////
            ;3985 ;
            ;3986
            ;3987 T7.CFBIT.OK3:
U 1318. 000D.2034.0180.0A40.0010.1319  ;3988  ALU_Q[AND]R[8],CLK.UBCC          ; Check SBI Fault Bit Set
U 1319. 0000.013C.0180.0800.0000.113E  ;3989  Z?
U 113E. 0000.003C.0180.0800.0000.131A  ;3990 =0 J/T7.SFBIT.OK3    ; OK .. SBI FAULT SET
```

```
            ;3991   .PAGE
            ;3992 ;
            ;3993 ;////////////////////////////////////////////////////////////////////////////////
            ;3994 ;
U 113F, 0000,003D,0980,0800,0084,70B0  ;3995     ERROR1[.2]    ; SBI Parity Fault CLEAR shoud
            ;3996 ;       ; ...  be SET
            ;3997 ;
            ;3998 ;
            ;3999 ;////////////////////////////////////////////////////////////////////////////////
            ;4000 ;
            ;4001 ; ERROR LOGOUT:
            ;4002 ;
            ;4003 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4004 ;  ID FAULT REGISTER
            ;4005 ;
            ;4006 ;////////////////////////////////////////////////////////////////////////////////
            ;4007 ;
            ;4008
            ;4009 T7.SFBIT.OK3:
U 131A, 0011,2034,01C0,0948,0010,131B  ;4010  Q_ALU,ALU_Q[AND]rc[09],CLK.UBCC ; Did CPU see Parity Fault ?
U 131B, 0001,203C,0180,09E8,0000,131C  ;4011  RC[0D]_Q    ; Save for error logout
U 131C, 0800,003C,0180,0A08,0000,131D  ;4012  D_R[1]     ; Set Parity Error Bit
U 131D, 080D,0030,0180,0A50,0000,131E  ;4013  D_ALU,ALU_D[OR]R[0A]  ; Was CPU the transmitter
U 131E, 001D,2020,0180,0800,0010,131F  ;4014  ALU_Q[XOR]D,CLK.UBCC  ; Did CPU see Parity Error and
U 131F, 0001,013C,0180,09E0,0000,1140  ;4015  Z?,RC[0C]_d    ; ...   as the Transmitter
```

```
            ;4016  .PAGE
            ;4017 ;
            ;4018 ;////////////////////////////////////////////////////////////////////////////////
            ;4019 ;
U 1140, 0000,003D,0D80,0800,0084,70B0  ;4020 =0 ERROR1[.3]    ; NO wrong bits set
            ;4021 ;       ; Either bit 31 or 26 in ID
            ;4022 ;       ; ... Fault not set
            ;4023 ;
            ;4024 ;////////////////////////////////////////////////////////////////////////////////
            ;4025 ;
            ;4026 ; ERROR LOGOUT:
            ;4027 ;
            ;4028 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4029 ;  EXPECTED SBI- FAULT- STATUS BITS (31-26)
            ;4030 ;  MASKED SBI-FAULT-STATUS BITS (31-26)
            ;4031 ;
            ;4032 ;////////////////////////////////////////////////////////////////////////////////
            ;4033 ;
            ;4034
U 1141, 0010,0038,0180,0970,0200,1320  ;4035  VA_RC[0E]    ; Did memory see fault ?
U 1320, 0000,003C,0180,C800,0000,1321  ;4036  D[LONG]_CACHE.P    ; Read Config A Reg
U 1321, 0811,0034,0180,0948,0000,1322  ;4037  D_ALU,ALU_D[AND]RC[09]  ; Mask fault bits
U 1322, 0001,003C,0180,09D8,0000,1323  ;4038  RC[0B]_D    ; Save masked image
U 1323, 0818,0038,3580,0800,0000,1324  ;4039  D_K[.50]    ; Generate the expected interrupt
U 1324, 0819,0030,8580,0800,0000,1325  ;4040  D_D.OR.K[.C]    ; ....vector
U 1325, 0001,003C,0180,09E8,0000,1044  ;4041  RC[0D]_D    ; Save expected interrupt vector
U 1044, 0000,003D,0180,0800,0000,1270  ;4042 =00 CALL[CHECK.VECTOR]  ; Was the correct Interrupt Vector
            ;4043       ; ...generated ?
U 1045, 0000,003C,0180,0800,0000,1326  ;4044  J/VECTOK.4    ; OK. Correct vector was generated.
```

```
            :4045  .PAGE
            :4046 ;
            :4047 ;////////////////////////////////////////////////////////////////////////////////
            :4048 ;
U 1046, 0000,003D,D980,0800,0084,70B0  :4049  ERROR1[.9]    ; Wrong vector was generated
            :4050
            :4051 ;
            :4052 ;////////////////////////////////////////////////////////////////////////////////
            :4053 ;
            :4054 ; ERROR LOGOUT:
            :4055 ;
            :4056 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            :4057 ;  EXPECTED INTERRUPT VECTOR
            :4058 ;  NOT USED
            :4059 ;  NOT USED
            :4060 ;  NOT USED
            :4061 ;  NOT USED
            :4062 ;  ID VECTOR REGISTER
            :4063 ;  ID SBI ERROR REGISTER
            :4064 ;  ID FAULT REGISTER
            :4065 ;
            :4066 ;
            :4067 ;////////////////////////////////////////////////////////////////////////////////
            :4068 ;
            :4069 =
            :4070 VECTOK.4:
U 1326, 0000,003C,01C0,0A08,0000,1327  :4071  Q_R[1]     ; Get parity fault mask
U 1327, 0011,2020,0180,0958,0010,1328  :4072  ALU_Q[XOR]RC[0B],CLK.UBCC ; Test memory sbi fault status
U 1328, 0001,213C,0180,09E0,0000,1142  :4073  Z?,RC[0C]_Q    ; Ckeck condition codes
```

```
            :4074  .PAGE
            :4075  ;
            :4076  ;////////////////////////////////////////////////////////////////////////////////
            :4077  ;
U 1142, 0000,003D,1180,0800,0084,70B0  ;4078 =0 ERROR1[.4]   ; Wrong Fault bits set
            :4079  ;
            :4080  ;
            :4081  ;////////////////////////////////////////////////////////////////////////////////
            :4082  ;
            :4083  ; ERROR LOGOUT:
            :4084  ;
            :4085  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            :4086  ;  NOT USED
            :4087  ;  EXPECTED CONTENTS OF CNFG A REGISTER
            :4088  ;  RECIEVED CONTENTS OF CNFG A REGISTER
            :4089  ;
            :4090  ;////////////////////////////////////////////////////////////////////////////////
            :4091  ;
U 1143, 0000,003C,0180,0800,0000,1144  ;4092  NOP
U 1144, 0000,003D,0180,0800,0000,10C0  ;4093 =0 CALL[SBI.INIT]   ; Clear the Sbi fault registers
U 1145, 0000,003C,6DF0,2C00,0000,1329  ;4094  Q_ID[FAULT]   ; ID Fault Register to Q
U 1329, 0001,203C,0180,09E8,0000,132A  ;4095  RC[0D]_Q   ; Save in RC[0D] for error logout
U 132A, 000D,2034,0180,0A40,0010,132B  ;4096  ALU_Q[AND]R[8],CLK.UBCC  ; SBI Fault Clear  ?
U 132B, 0000,013C,0180,0800,0000,1146  ;4097  Z?    ; Check condition codes
```

```
            :4098  .PAGE
            :4099 ;
            :4100 ;////////////////////////////////////////////////////////////////////////////////
            :4101 ;
U 1146, 0000,003D,0980,0800,0084,70B0  :4102 =0 ERROR1[.2]    ; NO SBI Fault Bit is set
            :4103 ;
            :4104 ;////////////////////////////////////////////////////////////////////////////////
            :4105 ;
            :4106 ; ERROR LOGOUT:
            :4107 ;
            :4108 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            :4109 ;  ID FAULT REGISTER
            :4110 ;
            :4111 ;////////////////////////////////////////////////////////////////////////////////
            :4112 ;
            :4113
U 1147, 000D,2034,0180,0A00,0010,132C  :4114  ALU_Q[AND]R[0],CLK.UBCC          ; CPU Fault Clear ?
U 132C, 0000,013C,0180,0800,0000,1148  :4115  Z?
```

```
            ;4116  .PAGE
            ;4117 ;
            ;4118 ;////////////////////////////////////////////////////////////////////////////////
            ;4119 ;
U 1148, 0000.003D.0980.0800.0084.70B0  ;4120 =0 ERROR1[.2]    ; NO CPU Fault did NOT clear
            ;4121 ;
            ;4122 ;
            ;4123 ;////////////////////////////////////////////////////////////////////////////////
            ;4124 ;
            ;4125 ; ERROR LOGOUT:
            ;4126 ;
            ;4127 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4128 ;  ID FAULT REGISTER
            ;4129 ;
            ;4130 ;////////////////////////////////////////////////////////////////////////////////
            ;4131 ;
U 1149, 0011.2034.01C0.0948.0000.132D  ;4132  Q_ALU,ALU_Q[AND]rc[09]  ;CPU SBI FAULT STATUS BITS CLEAR ?
U 132D, 0001.203C.0180.09E8.0010.132E  ;4133  RC[0D]_Q,CLK.UBCC
U 132E, 0003.013C.0180.09E0.0000.114A  ;4134          Z?,RC[0C]_0
```

```
            :4135   .PAGE
            :4136   ;
            :4137   ;////////////////////////////////////////////////////////////////////////////////
            :4138   ;
U 114A. 0000.003D.0180.0800.0000.11E5   ;4139 =0 ERROR1     ;CPU SBI-STATUS DID NOT CLEAR
            :4140   ;
            :4141   ;////////////////////////////////////////////////////////////////////////////////
            :4142   ;
            :4143   ; ERROR LOGOUT:
            :4144   ;
            :4145   ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            :4146   ;  MASKED SBI-FAULT STATUS BITS READ BITS (31-26)
            :4147   ;  EXPECTED SBI_FAULT STATUS BITS (31-26)
            :4148   ;
            :4149   ;////////////////////////////////////////////////////////////////////////////////
            :4150   ;
U 114B. 0010.0038.0180.0970.0200.132F   ;4151   VA_RC[0E]
U 132F. 0000.003C.0180.C800.0000.1330   ;4152   D[LONG]_CACHE.P    ; Read CNFG A save in  D Register
U 1330. 0811.0034.0180.0948.0000.1331   ;4153   D_ALU,ALU_D[AND]rc[09]   ; Mask Fault Bits
U 1331. 0001.003C.0180.09E8.0000.1332   ;4154   RC[0D]_D    ; Save Masked Bits
U 1332. 0810.0038.0180.0968.0010.1333   ;4155   D_RC[0D],CLK.UBCC
U 1333. 0000.013C.0180.0800.0000.114C   ;4156   Z?
```

```
            ;4157  .PAGE
            ;4158 ;
            ;4159 ;////////////////////////////////////////////////////////////////////////////
            ;4160 ;
U 114C. 0000.003D.0D80.0800.0084.70B0  ;4161 =0 ERROR1[.3]    ; SBI Fault Status Bits set
            ;4162 ;      ; ... should be cleared
            ;4163 ;
            ;4164 ;////////////////////////////////////////////////////////////////////////////
            ;4165 ;
            ;4166 ; ERROR LOGOUT:
            ;4167 ;
            ;4168 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4169 ;  MASKED FAULTS BITS FROM CNFG A REGISTER
            ;4170 ;  EXPECTED FAULTS BITS
            ;4171 ;
            ;4172 ;////////////////////////////////////////////////////////////////////////////
            ;4173 ;
            ;4174 ;  SUBTEST 4 ** WRITE DATA SEQUENCE FAULT **
            ;4175 ;
            ;4176 ;
            ;4177 ;////////////////////////////////////////////////////////////////////////////
            ;4178 ;
U 114D. 0000.003C.0180.0800.0000.114E  ;4179  NOP
U 114E. 0000.003D.0180.0800.0000.11F0  ;4180 =0 SUBTEST    ; Increment the subtest number
U 114F. 0000.003C.0180.0800.0000.1150  ;4181  NOP     ; ...
U 1150. 0000.003D.0180.0800.0000.11E4  ;4182 =0 ERLOOP     ; Set U address for error looping
U 1151. 0000.003C.0180.0800.0000.1152  ;4183  NOP     ; ...
U 1152. 0000.003D.0180.0800.0000.1283  ;4184 =0 CALL[INIT]    ; Initialize the SBI
U 1153. 0800.003C.0180.0A48.0000.1334  ;4185  D_R[9]     ; Generate cache off,force
U 1334. 080D.0030.0180.0A18.0000.1335  ;4186  D_ALU,ALU_D[OR]R[3]  ; ...  write sequence fault in D Reg
U 1335. 0000.003C.7580.3C00.0000.1336  ;4187  ID[MAINT]_D    ; Set the Bit in the ID maint register
U 1336. 0F00.003C.0180.0800.0000.1337  ;4188  D_0     ; Point to location 0
U 1337. 0001.003C.0180.0800.0200.1338  ;4189  VA_D     ; ...in cache
U 1338. 0000.003C.0180.A800.0000.1339  ;4190  CACHE.P_D[LONG]    ; Force the error
U 1339. 0000.003C.0180.0800.0000.133A  ;4191  NOP     ; Wait for the SBI to settle
U 133A. 0000.003C.0180.0800.0000.133B  ;4192  NOP     ; ...
U 133B. 0000.003C.0180.0800.0000.133C  ;4193  NOP     ;CLOCK TICS
U 133C. 0000.003C.0180.0800.0000.133D  ;4194  NOP     ; ...
U 133D. 0000.003C.0180.0800.0000.133E  ;4195  NOP     ; ...
U 133E. 0000.003C.0180.0800.0000.133F  ;4196  NOP     ; ...
U 133F. 0800.003C.0180.0A48.0000.1340  ;4197  D_R[9]     ; Clear force fault bit
U 1340. 0000.003C.7580.3C00.0000.1154  ;4198  ID[MAINT]_D
            ;4199 =0 D_K[.FF],    ; Set up number of cycles to wait
U 1154. 0818.0039.4980.0800.0000.1219  ;4200   CALL[WAIT.LOOP]  ; Wait for SBI to settle
U 1155. 0818.0038.3580.0800.0000.1341  ;4201  D_K[.50]    ; Generate the expected interrupt
U 1341. 0819.0030.8580.0800.0000.1342  ;4202  D_D.OR.K[.C]    ; ....vector
U 1342. 0001.003C.0180.09E8.0000.1048  ;4203  RC[0D]_D    ; Save in RC[0D] for error logout
U 1048. 0000.003D.0180.0800.0000.1270  ;4204 =00 CALL[CHECK.VECTOR]  ; Was the correct Interrupt Vector
            ;4205      ; ...generated ?
U 1049. 0000.003C.0180.0800.0000.1343  ;4206  J/VECTOK.5    ; OK. Correct vector was generated
```

```
            ;4207  .PAGE
            ;4208 ;
            ;4209 ;////////////////////////////////////////////////////////////////////////////////
            ;4210 ;
U 104A, 0000,003D,D980,0800,0084,70B0  ;4211  ERROR1[.9]   ; Wrong vector was generated
            ;4212
            ;4213 ;
            ;4214 ;////////////////////////////////////////////////////////////////////////////////
            ;4215 ;
            ;4216 ; ERROR LOGOUT:
            ;4217 ;
            ;4218 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4219 ;  EXPECTED INTERRUPT VECTOR
            ;4220 ;  NOT USED
            ;4221 ;  NOT USED
            ;4222 ;  NOT USED
            ;4223 ;  NOT USED
            ;4224 ;  ID VECTOR REGISTER
            ;4225 ;  ID SBI ERROR REGISTER
            ;4226 ;  ID FAULT REGISTER
            ;4227 ;
            ;4228 ;
            ;4229 ;////////////////////////////////////////////////////////////////////////////////
            ;4230 ;
            ;4231 =
            ;4232 VECTOK.5:
U 1343, 0000,003C,6DF0,2C00,0000,1344  ;4233  Q_ID[FAULT]   ; Check CPU Fault Bit Set
U 1344, 0001,203C,0180,09E8,0000,1345  ;4234  RC[0D]_Q   ; Save for error logout
U 1345, 000D,2034,0180,0A00,0010,1346  ;4235  ALU_Q[AND]R[0],CLK.UBCC
U 1346, 0000,013C,0180,0800,0000,1156  ;4236  Z?
U 1156, 0000,003C,0180,0800,0000,1347  ;4237 =0 J/T7.CFBIT.OK4   ; O.K. BIT SET
```

```
            ;4238  .PAGE
            ;4239 ;
            ;4240 ;////////////////////////////////////////////////////////////////////////////////
            ;4241 ;
U 1157. 0000.003D.0980.0800.0084.70B0  ;4242   ERROR1[.2]     ; CPU Fault was clear
            ;4243 ;       ; ...should be set
            ;4244 ;
            ;4245 ;////////////////////////////////////////////////////////////////////////////////
            ;4246 ;
            ;4247 ; ERROR LOGOUT:
            ;4248 ;
            ;4249 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4250 ;  ID FAULT REGISTER
            ;4251 ;
            ;4252 ;////////////////////////////////////////////////////////////////////////////////
            ;4253 ;
            ;4254
            ;4255 T7.CFBIT.OK4:
U 1347. 000D.2034.0180.0A40.0010.1348  ;4256  ALU_Q[AND]R[8],CLK.UBCC          ; Check SBI Fault Bit Set
U 1348. 0000.013C.0180.0800.0000.1158  ;4257  Z?
U 1158. 0000.003C.0180.0800.0000.1349  ;4258 =0 J/T7.SFBIT.OK4    ; OK .. SBI FAULT SET
```

```
            ;4259  .PAGE
            ;4260 ;
            ;4261 ;////////////////////////////////////////////////////////////////////////////////
            ;4262 ;
U 1159, 0000.003D.0980.0800.0084.70B0  ;4263     ERROR1[.2]     ; SBI Fault CLEAR shoud
            ;4264 ;       ; ...  be SET
            ;4265 ;
            ;4266 ;
            ;4267 ;////////////////////////////////////////////////////////////////////////////////
            ;4268 ;
            ;4269 ; ERROR LOGOUT:
            ;4270 ;
            ;4271 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4272 ;  ID FAULT REGISTER
            ;4273 ;
            ;4274 ;////////////////////////////////////////////////////////////////////////////////
            ;4275 ;
            ;4276 T7.SFBIT.OK4:
U 1349, 0011.2034.01C0.0948.0000.134A  ;4277  Q_ALU,ALU_Q[AND]rc[09]     ; Check Cpu SBI=STATUS bits
U 134A, 0001.203C.0180.09E8.0000.134B  ;4278  RC[0D]_Q    ; Save for error logout
U 134B, 000D.2020.0180.0A50.0010.134C  ;4279  ALU_Q[XOR]R[0A],CLK.UBCC ; Did CPU see Parity Error and
U 134C, 0001.013C.0180.09E0.0000.115A  ;4280  Z?,RC[0C]_d    ; ...    as the Transmitter
```

```
            ;4281   .PAGE
            ;4282 ;
            ;4283 ;////////////////////////////////////////////////////////////////////////////////
            ;4284 ;
U 115A, 0000,003D,0D80,0800,0084,70B0  ;4285 =0 ERROR1[.3]     ; NO wrong bits set
            ;4286 ;       ; Bit 26 in ID
            ;4287 ;       ; ... Fault not set
            ;4288 ;
            ;4289 ;////////////////////////////////////////////////////////////////////////////////
            ;4290 ;
            ;4291 ; ERROR LOGOUT:
            ;4292 ;
            ;4293 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4294 ;  EXPECTED SBI- FAULT- STATUS BITS (31-26)
            ;4295 ;  MASKED SBI-FAULT-STATUS BITS (31-26)
            ;4296 ;
            ;4297 ;////////////////////////////////////////////////////////////////////////////////
            ;4298 ;
U 115B, 0010,0038,0180,0970,0200,134D  ;4299  VA_RC[0E]    ; Did memory see fault ?
U 134D, 0000,003C,0180,C800,0000,134E  ;4300  D[LONG]_CACHE.P    ; Read Config A Reg
U 134E, 0811,0034,0180,0948,0000,134F  ;4301  D_ALU,ALU_D[AND]RC[09]   ; Mask fault bits
U 134F, 0001,003C,0180,09D8,0000,1350  ;4302  RC[0B]_D    ; Save masked image
U 1350, 0818,0038,3580,0800,0000,1351  ;4303  D_K[.50]    ; Generate the expected interrupt
U 1351, 0819,0030,8580,0800,0000,1352  ;4304  D_D.OR.K[.C]    ; ....vector
U 1352, 0001,003C,0180,09E8,0000,1050  ;4305  RC[0D]_D    ; Save in RC[0D] for error logout
U 1050, 0000,003D,0180,0800,0000,1270  ;4306 =00 CALL[CHECK.VECTOR]   ; Was the correct Interrupt Vector
            ;4307       ; ...generated ?
U 1051, 0000,003C,0180,0800,0000,1353  ;4308  J/VECTOK.6    ; OK. Correct vector was generated
```

```
            ;4309  .PAGE
            ;4310 ;
            ;4311 ;////////////////////////////////////////////////////////////////////////////////
            ;4312 ;
U 1052. 0000.003D.D980.0800.0084.70B0  ;4313  ERROR1[.9]   ; Wrong vector was generated
            ;4314
            ;4315 ;
            ;4316 ;////////////////////////////////////////////////////////////////////////////////
            ;4317 ;
            ;4318 ; ERROR LOGOUT:
            ;4319 ;
            ;4320 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4321 ;  EXPECTED INTERRUPT VECTOR
            ;4322 ;  NOT USED
            ;4323 ;  NOT USED
            ;4324 ;  NOT USED
            ;4325 ;  NOT USED
            ;4326 ;  ID VECTOR REGISTER
            ;4327 ;  ID SBI ERROR REGISTER
            ;4328 ;  ID FAULT REGISTER
            ;4329 ;
            ;4330 ;
            ;4331 ;////////////////////////////////////////////////////////////////////////////////
            ;4332 ;
            ;4333 =
            ;4334 VECTOK.6:
U 1353. 0800.003C.0180.0A30.0000.1354  ;4335  d_R[6]    ; Get write sequence fault mask
U 1354. 0011.0020.0180.0958.0010.1355  ;4336  ALU_D[XOR]RC[0B],CLK.UBCC ; Test for only write sequence  fault
U 1355. 0001.013C.0180.09E0.0000.115C  ;4337  Z?,RC[0C]_D   ; Ckeck condition codes
```

```
            ;4338  .PAGE
            ;4339 ;
            ;4340 ;////////////////////////////////////////////////////////////////////////////////
            ;4341 ;
U 115C. 0000.003D.1180.0800.0084.70B0  ;4342 =0 ERROR1[.4]   ; Wrong Fault bits set
            ;4343 ;
            ;4344 ;
            ;4345 ;////////////////////////////////////////////////////////////////////////////////
            ;4346 ;
            ;4347 ; ERROR LOGOUT:
            ;4348 ;
            ;4349 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4350 ;  NOT USED
            ;4351 ;  EXPECTED CONTENTS OF CNFG A REGISTER
            ;4352 ;  RECIEVED CONTENTS OF CNFG A REGISTER
            ;4353 ;
            ;4354 ;////////////////////////////////////////////////////////////////////////////////
            ;4355 ;
U 115D. 0000.003C.0180.0800.0000.115E  ;4356  NOP
U 115E. 0000.003D.0180.0800.0000.10C0  ;4357 =0 CALL[SBI.INIT]   ; Clear the Sbi fault registers
U 115F. 0000.003C.6DF0.2C00.0000.1356  ;4358  Q_ID[FAULT]   ; ID Fault Register to Q
U 1356. 0001.203C.0180.09E8.0000.1357  ;4359  RC[0D]_Q   ; Save in RC[0D] for error logout
U 1357. 000D.2034.0180.0A40.0010.1358  ;4360  ALU_Q[AND]R[8],CLK.UBCC  ; SBI Fault Clear  ?
U 1358. 0000.013C.0180.0800.0000.1160  ;4361  Z?    ; Check condition codes
```

```
            ;4362  .PAGE
            ;4363 ;
            ;4364 ;////////////////////////////////////////////////////////////////////////////
            ;4365 ;
U 1160. 0000.003D.0980.0800.0084.70B0  ;4366 =0 ERROR1[.2]     ; NO SBI Fault Bit is set
            ;4367 ;
            ;4368 ;////////////////////////////////////////////////////////////////////////////
            ;4369 ;
            ;4370 ; ERROR LOGOUT:
            ;4371 ;
            ;4372 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4373 ;  ID FAULT REGISTER
            ;4374 ;
            ;4375 ;////////////////////////////////////////////////////////////////////////////
            ;4376 ;
            ;4377
U 1161. 000D.2034.0180.0A00.0010.1359  ;4378  ALU_Q[AND]R[0].CLK.UBCC          ; CPU Fault Clear ?
U 1359. 0000.013C.0180.0800.0000.1162  ;4379  Z?
```

```
            ;4380  .PAGE
            ;4381 ;
            ;4382 ;////////////////////////////////////////////////////////////////////////////////
            ;4383 ;
U 1162. 0000.003D.0980.0800.0084.70B0  ;4384 =0 ERROR1[.2]    ; NO CPU Fault did NOT clear
            ;4385 ;
            ;4386 ;
            ;4387 ;////////////////////////////////////////////////////////////////////////////////
            ;4388 ;
            ;4389 ; ERROR LOGOUT:
            ;4390 ;
            ;4391 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4392 ;  ID FAULT REGISTER
            ;4393 ;
            ;4394 ;////////////////////////////////////////////////////////////////////////////////
            ;4395 ;
U 1163. 0011.2034.01C0.0948.0000.135A  ;4396  Q_ALU,ALU_Q[AND]rc[09]  ;CPU SBI FAULT STATUS BITS CLEAR ?
U 135A. 0001.203C.0180.09E8.0010.135B  ;4397  RC[0D]_Q,CLK.UBCC
U 135B. 0003.013C.0180.09E0.0000.1164  ;4398          Z?,RC[0C]_0
```

```
            ;4399  .PAGE
            ;4400 ;
            ;4401 ;////////////////////////////////////////////////////////////////////////////
            ;4402 ;
U 1164. 0000.003D.0D80.0800.0084.70B0  ;4403 =0 ERROR1[.3]   ;CPU SBI-STATUS DID NOT CLEAR
            ;4404 ;
            ;4405 ;////////////////////////////////////////////////////////////////////////////
            ;4406 ;
            ;4407 ; ERROR LOGOUT:
            ;4408 ;
            ;4409 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4410 ;  MASKED SBI-FAULT STATUS BITS READ BITS (31-26)
            ;4411 ;  EXPECTED SBI_FAULT STATUS BITS (31-26)
            ;4412 ;
            ;4413 ;////////////////////////////////////////////////////////////////////////////
            ;4414 ;
U 1165. 0010.0038.0180.0970.0200.135C  ;4415  VA_RC[0E]
U 135C. 0000.003C.0180.C800.0000.135D  ;4416  D[LONG]_CACHE.P   ; Read CNFG A save in  D Register
U 135D. 0811.0034.0180.0948.0000.135E  ;4417  D_ALU,ALU_D[AND]rc[09]   ; Mask Fault Bits
U 135E. 0001.003C.0180.09E8.0000.135F  ;4418  RC[0D]_D   ; Save Masked Bits
U 135F. 0810.0038.0180.0968.0010.1360  ;4419  D_RC[0D],CLK.UBCC
U 1360. 0000.013C.0180.0800.0000.1166  ;4420  Z?
```

```
            ;4421  .PAGE
            ;4422 ;
            ;4423 ;////////////////////////////////////////////////////////////////////////////////
            ;4424 ;
U 1166. 0000.003D.0D80.0800.0084.70B0  ;4425 =0 ERROR1[.3]    ; SBI Fault Status Bits set
            ;4426 ;      ; ... should be cleared
            ;4427 ;
            ;4428 ;////////////////////////////////////////////////////////////////////////////////
            ;4429 ;
            ;4430 ;** SUBTEST 5 -NORMAL INTERLOCK SEQUENCE -
            ;4431 ;
            ;4432 ;
            ;4433 ;         CHECK THE NORMAL INTERLOCK SEQUENCE OF OPERATIONS
            ;4434 ;  I.E. INTERLOCK READ,INTERLOCK  WRITE
            ;4435 ;  CHECKS THE OPERATIONS WITH COMPLIMENT DATA.
            ;4436 ;
            ;4437 ;////////////////////////////////////////////////////////////////////////////////
            ;4438 ;
U 1167. 0000.003C.0180.0800.0000.1168  ;4439  NOP
U 1168. 0000.003D.0180.0800.0000.11F0  ;4440 =0 SUBTEST    ; Increment the subtest number
U 1169. 0000.003C.0180.0800.0000.116A  ;4441  NOP     ; ...
U 116A. 0000.003D.0180.0800.0000.11E4  ;4442 =0 ERLOOP    ; Set U address for error loop
U 116B. 0000.003C.0180.0800.0000.116C  ;4443  NOP     ; ...
U 116C. 0000.003D.0180.0800.0000.1283  ;4444 =0 CALL[INIT]   ; Initialize the SBI
U 116D. 0F00.003C.0180.0800.0000.1361  ;4445  D_0    ; Point to location 0
U 1361. 0001.003C.0180.0800.0200.116E  ;4446  VA_D    ; ...in the VA
U 116E. 0800.003C.0180.0A68.0000.116F  ;4447 =0 D_R[0D]    ; Get data to initialize memory
U 116F. 0001.003C.0180.09E0.0000.1362  ;4448  RC[0C]_D   ; Save data for logout
U 1362. 0000.003C.0180.A800.0000.1170  ;4449  CACHE.P_D[LONG]   ; Initialize the location
            ;4450 =0 D_K[.FF],    ; Get number of cycles to wait
U 1170. 0818.0039.4980.0800.0000.1219  ;4451   CALL[WAIT.LOOP]  ; Wait for SBI to finish write cycle
U 1171. 0000.003C.0180.0800.0000.1054  ;4452  NOP     ; ...
U 1054. 0000.003D.0180.0800.0000.121A  ;4453 =00 CHECK.INTERRUPT    ; Check for unexpected interrupts
U 1055. 0000.003C.0180.0800.0000.1172  ;4454  J/T7.S5.NOINT1    ; Jump if no interrupts
```

```
          ;4455  .PAGE
          ;4456 ;
          ;4457 ;////////////////////////////////////////////////////////////////////////////////
          ;4458 ;
U 1056. 0000.003D.F580.0800.0084.70B0  ;4459  ERROR1[.A]    ; !!!!UNEXPECTED INTERRUPT!!!!
          ;4460 ;
          ;4461 ;
          ;4462 ;////////////////////////////////////////////////////////////////////////////////
          ;4463 ;
          ;4464 ; ERROR LOGOUT:
          ;4465 ;
          ;4466 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
          ;4467 ;  NOT USED
          ;4468 ;  NOT USED
          ;4469 ;  NOT USED
          ;4470 ;  NOT USED
          ;4471 ;  NOT USED
          ;4472 ;  INTERRUPT VECTOR REGISTER
          ;4473 ;  SBI.ERROR REGISTER
          ;4474 ;  SBI.FAULT REGISTER
          ;4475 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
          ;4476 ;
          ;4477 ;////////////////////////////////////////////////////////////////////////////////
          ;4478 =
          ;4479 =0
          ;4480 T7.S5.NOINT1:
U 1172. 0000.003D.8980.0800.0084.7231  ;4481  SET.TRAP.MASK[.D]  ; Enable Time-Out traps
U 1173. 0000.003C.0180.E800.0000.1363  ;4482  MCT/LOCKREAD.P    ; Execute an Interlock Read
U 1363. 0001.003C.0180.09E8.0000.1060  ;4483  RC[0D]_D    ; Get data read from memory
U 1060. 0000.003D.0180.0800.0000.1227  ;4484 =00 CHECK.UTRAP    ; Any  Time-Out U-Traps occur ?
U 1061. 0000.003C.0180.0800.0000.1064  ;4485  J/T7.S5.NOTIMO1    ; Jump if no timeout
```

```
            ;4486  .PAGE
            ;4487 ;
            ;4488 ;////////////////////////////////////////////////////////////////////////////////
            ;4489 ;
U 1062. 0000.003D.0180.0800.0084.70B0  ;4490  ERROR1[.8]    ;INTERLOCK READ TIMED OUT
            ;4491 ;
            ;4492 ;
            ;4493 ;////////////////////////////////////////////////////////////////////////////////
            ;4494 ;
            ;4495 ; ERROR LOGOUT:
            ;4496 ;
            ;4497 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4498 ;  RECIEVED DATA FROM MEMORY
            ;4499 ;  EXPECTED DATA FROM MEMORY
            ;4500 ;  NOT USED
            ;4501 ;  NOT USED
            ;4502 ;  NOT USED
            ;4503 ;  EXPECTED U TPAP FLAG
            ;4504 ;  RECIEVED U TRAP FLAG
            ;4505 ;
            ;4506 ;////////////////////////////////////////////////////////////////////////////////
            ;4507 ;
            ;4508 =
            ;4509 =00
            ;4510 T7.S5.NOTIMO1:
U 1064. 0000.003D.0180.0800.0000.121A  ;4511  CHECK.INTERRUPT   ; Check for unexpected interrupts
U 1065. 0000.003C.0180.0800.0000.1364  ;4512  J/T7.S5.NOINT2   ; Jump if no interrupts
```

```
            ;4513  .PAGE
            ;4514 ;
            ;4515 ;////////////////////////////////////////////////////////////////////////////
            ;4516 ;
U 1066, 0000,003D,F580,0800,0084,70B0  ;4517  ERROR1[.A]   ; !!! UNEXPECTED INTERRUPT !!!
            ;4518 ;
            ;4519 ;
            ;4520 ;////////////////////////////////////////////////////////////////////////////
            ;4521 ;
            ;4522 ; ERROR LOGOUT:
            ;4523 ;
            ;4524 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4525 ;  RECIVIED DATA FROM MEMORY
            ;4526 ;  EXPECTED DATA FROM MEMORY
            ;4527 ;  NOT USED
            ;4528 ;  NOT USED
            ;4529 ;  NOT USED
            ;4530 ;  INTERRUPT VECTOR REGISTER
            ;4531 ;  SBI.ERROR REGISTER
            ;4532 ;  SBI.FAULT REGISTER
            ;4533 ;  INTERRUPT OCCCURED FLAG (TRUE=1, FALSE=0)
            ;4534 ;
            ;4535 ;////////////////////////////////////////////////////////////////////////////
            ;4536 ;
            ;4537 =
            ;4538 T7.S5.NOINT2:
U 1364, 0810,0038,0180,0968,0000,1365  ;4539  D_RC[0D]   ; Put saved data from memory back in D
U 1365, 000D,0020,0180,0A68,0010,1366  ;4540  ALU_D[XOR]R[0D],CLK.UBCC ; Compare the data read
U 1366, 0000,013C,0180,0800,0000,1174  ;4541  Z?    ; Check condition codes
```

```
            ;4542  .PAGE
            ;4543 ;
            ;4544 ;////////////////////////////////////////////////////////////////////////////
            ;4545 ;
U 1174, 0000.003D.0D80.0800.0084.70B0  ;4546 =0 ERROR1[.3]    ; !!!INTERLOCK READ FAILED!!!
            ;4547 ;
            ;4548 ;
            ;4549 ;////////////////////////////////////////////////////////////////////////////
            ;4550 ;
            ;4551 ; ERROR LOGOUT:
            ;4552 ;
            ;4553 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4554 ;  RECIEVED  DATA FROM MEMORY
            ;4555 ;  EXPECTED  DATA FROM MEMORY
            ;4556 ;
            ;4557 ;////////////////////////////////////////////////////////////////////////////
            ;4558 ;
U 1175, 0000.003C.0180.0800.0000.1176  ;4559  NOP
U 1176, 0000.003D.0180.0800.0000.1283  ;4560 =0 CALL[INIT]   ; Initialize the Sbi
U 1177, 0018.0038.1980.0800.0200.1367  ;4561  VA_K[ZERO]   ; Point to location 0
U 1367, 0000.003C.0180.E800.0000.1368  ;4562  MCT/LOCKREAD.P   ; Do a Interlock read
U 1368, 0810.0038.0180.0910.0000.1369  ;4563  D_RC[02]   ; Load compliment data for INT Write
U 1369, 0001.003C.0180.09E0.0000.136A  ;4564  RC[0C]_D   ; Save expected data
U 136A, 0000.003C.0180.B800.0000.1178  ;4565  MCT/LOCKWRITE.P   ; Execute an Interlock Write
            ;4566 =0 D_K[.FF],   ; Get number of cycles to wait
U 1178, 0818.0039.4980.0800.0000.1219  ;4567   CALL[WAIT.LOOP]  ; Wait for SBI to finish write cycle
U 1179, 0000.003C.0180.0800.0000.1068  ;4568  NOP    ; ...
U 1068, 0000.003D.8580.0800.0084.722D  ;4569 =00 CHECK.SBI.ERR[.C]  ; See if there was an SBI timeout
U 1069, 0000.003C.0180.0800.0000.106C  ;4570  J/T7.S5.NOTIMO2   ; Jump if NO timeout
```

```
            ;4571   .PAGE
            ;4572 ;
            ;4573 ;////////////////////////////////////////////////////////////////////////////////
            ;4574 ;
U 106A, 0000,003D,5D80,0800,0084,70B0  ;4575  ERROR1[.7]    ; !!!INTERLOCK WRITE TIMEOUT!!!
            ;4576 ;
            ;4577 ;
            ;4578 ;////////////////////////////////////////////////////////////////////////////////
            ;4579 ;
            ;4580 ; ERROR LOGOUT:
            ;4581 ;
            ;4582 ; DATA: I/O BASE ADDRESS OF MS780E CURRENTLY UNDER TEST
            ;4583 ;  NOT USED
            ;4584 ;  DATA TO BE WRITTEN
            ;4585 ;  NOT USED
            ;4586 ;  NOT USED
            ;4587 ;  NOT USED
            ;4588 ;  SBI.ERROR REGISTER
            ;4589 ;
            ;4590 ;////////////////////////////////////////////////////////////////////////////////
            ;4591 ;
            ;4592 =
            ;4593 =00
            ;4594 T7.S5.NOTIM02:
U 106C, 0000,003D,0180,0800,0000,121A  ;4595  CHECK.INTERRUPT    : Check for interrupts
U 106D, 0000,003C,0180,0800,0000,136B  ;4596  J/T7.S5.NOINT4    ; Jump if no interrupt
```

```
            ;4597  .PAGE
            ;4598 ;
            ;4599 ;////////////////////////////////////////////////////////////////////////////////
            ;4600 ;
U 106E, 0000,003D,F580,0800,0084,70B0  ;4601  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!
            ;4602 ;
            ;4603 ;
            ;4604 ;////////////////////////////////////////////////////////////////////////////////
            ;4605 ;
            ;4606 ; ERROR LOGOUT:
            ;4607 ;
            ;4608 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4609 ;  NOT USED
            ;4610 ;  DATA TO BE WRITTEN
            ;4611 ;  NOT USED
            ;4612 ;  NOT USED
            ;4613 ;  NOT USED
            ;4614 ;  INTERRUPT VECTOR REGISTER
            ;4615 ;  SBI.ERROR REGISTER
            ;4616 ;  SBI.FAULT REGISTER
            ;4617 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;4618 ;
            ;4619 ;////////////////////////////////////////////////////////////////////////////////
            ;4620 ;
            ;4621 =
            ;4622 T7.S5.NOINT4:
U 136B, 0000,003C,0180,C800,0000,136C  ;4623  MCT/READ.P   ; Read and check data written
            ;4624      ; ...with Interlock Write
U 136C, 0001,003C,0180,09E8,0000,136D  ;4625  RC[0D]_D   ; Save data read for error logout
U 136D, 0011,0020,0180,0910,0010,136E  ;4626  ALU_D[XOR]RC[02],CLK.UBCC ; Compare data
U 136E, 0000,013C,0180,0800,0000,117A  ;4627  Z?    ; Check the condition codes
```

```
            ;4628  .PAGE
            ;4629  ;
            ;4630  ;////////////////////////////////////////////////////////////////////////////////
            ;4631  ;
U 117A. 0000.003D.0D80.0800.0084.70B0  ;4632 =0 ERROR1[.3]    ; !!!INTERLOCK WRITE FAILED!!!
            ;4633  ;
            ;4634  ;
            ;4635  ;////////////////////////////////////////////////////////////////////////////////
            ;4636  ;
            ;4637  ; ERROR LOGOUT:
            ;4638  ;
            ;4639  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4640  ;  RECIEVED DATA FROM MEMORY
            ;4641  ;  EXPECTED DATA FROM MEMORY
            ;4642  ;
            ;4643  ;////////////////////////////////////////////////////////////////////////////////
            ;4644  ;
U 117B. 0000.003C.0180.0800.0000.117C  ;4645  NOP
U 117C. 0000.003D.8980.0800.0084.7231  ;4646 =0 SET.TRAP.MASK[.D]  ; Enable time-outs
U 117D. 0000.003C.0180.E800.0000.136F  ;4647  MCT/LOCKREAD.P   ; Execute another Interlock Read
            ;4648       ; ... to check that the interlockd
            ;4649       ; ... Flip Flop was cleared bt the
            ;4650       ; ... Interlockd write
U 136F. 0001.003C.0180.0AE8.0000.1070  ;4651  R[0D]_D    ; Save data read for error logout
U 1070. 0000.003D.0180.0800.0000.1227  ;4652 =00 CHECK.UTRAP   ; See if the time-out U trap occured
U 1071. 0000.003C.0180.0800.0000.1073  ;4653  J/T7.S5.NOTM03   ; Jump if no timeout
```

```
            ;4654  .PAGE
            ;4655 ;
            ;4656 ;////////////////////////////////////////////////////////////////////////////////
            ;4657 ;
U 1072, 0000,003D,0180,0800,0084,70B0  ;4658  ERROR1[.8]   ; !!!INTERLOCK READ TIMED OUT!!!
            ;4659       ; ... MEMORY INTERLOCK FLIP FLOP
            ;4660       ; ... NOT CLEARED
            ;4661 ;
            ;4662 ;////////////////////////////////////////////////////////////////////////////////
            ;4663 ;
            ;4664 ; ERROR LOGOUT:
            ;4665 ;
            ;4666 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4667 ;  NOT USED
            ;4668 ;  NOT USED
            ;4669 ;  NOT USED
            ;4670 ;  NOT USED
            ;4671 ;  NOT USED
            ;4672 ;  EXPECTED UTRAP FLAG
            ;4673 ;  RECIEVED UTRAP FLAG
            ;4674 ;
            ;4675 ;////////////////////////////////////////////////////////////////////////////////
            ;4676 ;
            ;4677 T7.S5.NOTM03:
U 1073, 0810,0038,0180,0968,0000,1370  ;4678  D_RC[0D]   ; Check the data read
U 1370, 0011,0020,0180,0960,0010,1371  ;4679 = ALU_D[XOR]RC[0C],CLK.UBCC ; ....
U 1371, 0000,013C,0180,0800,0000,117E  ;4680  Z?    ; Check condition codes
```

```
            ;4681   .PAGE
            ;4682 ;
            ;4683 ;////////////////////////////////////////////////////////////////////////////
            ;4684 ;
U 117E, 0000,003D,0D80,0800,0084,70B0  ;4685 =0 ERROR1[.3]    ; Data missed compared
            ;4686 ;
            ;4687 ;
            ;4688 ;////////////////////////////////////////////////////////////////////////////
            ;4689 ;
            ;4690 ; ERROR LOGOUT:
            ;4691 ;
            ;4692 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4693 ;  RECIEVED DATA FROM MEMORY
            ;4694 ;  EXPECTED DATA FROM MEMORY
            ;4695 ;
            ;4696 ;////////////////////////////////////////////////////////////////////////////
            ;4697 ;
U 117F, 0818,0038,4980,0800,0000,1180  ;4698  D_K[.FF]    ; Prepare to wait 256 clock ticks
U 1180, 0000,003D,0180,0800,0000,1219  ;4699 =0 CALL[WAIT.LOOP]    ; ... so that the Interlocked Flip
            ;4700       ; ... Flop will clear
U 1181, 0000,003C,0180,0800,0000,1182  ;4701  NOP     ; ....
            ;4702 ;
            ;4703 ;////////////////////////////////////////////////////////////////////////////
            ;4704 ;
            ;4705 ;** SUBTEST 6 ** INTERLOCK SEQUENCE FAULT
            ;4706 ;
            ;4707 ;////////////////////////////////////////////////////////////////////////////
            ;4708 ;
U 1182, 0000,003D,0180,0800,0000,11F0  ;4709 =0 SUBTEST     ; Incement the subtest number
U 1183, 0000,003C,0180,0800,0000,1184  ;4710  NOP     ; ...
U 1184, 0000,003D,0180,0800,0000,11E4  ;4711 =0 ERLOOP     ; Set U address for error loop
U 1185, 0000,003C,0180,0800,0000,1186  ;4712  NOP     ; ....
U 1186, 0000,003D,0180,0800,0000,1283  ;4713 =0 CALL[INIT]    ; Initialize the Sbi
U 1187, 0F00,003C,0180,0800,0000,1372  ;4714  D_0     ; Point to location 0
U 1372, 0001,003C,0180,0800,0200,1188  ;4715  VA_D     ; ...in the VA register
            ;4716 =0 D_K[.FF],         ; Set up number of cycles to wait
U 1188, 0818,0039,4980,0800,0000,1219  ;4717   CALL[WAIT.LOOP]  ; Wait for SBI to finish write cycle
U 1189, 0000,003C,0180,0800,0000,118A  ;4718  NOP     ; ...
U 118A, 0000,003D,8980,0800,0084,7231  ;4719 =0 SET.TRAP.MASK[.D]  ; Enable time-out traps
U 118B, 0000,003C,0180,B800,0000,118C  ;4720  MCT/LOCKWRITE.P    ; Initiate write operation
            ;4721 =0 D_K[.3FF].RIGHT,  ; Set up number of cycles to wait
U 118C, 0858,0039,8180,0800,0000,1219  ;4722   CALL[WAIT.LOOP]  ; Wait for SBI to finish write cycle
U 118D, 0818,0038,A580,0800,0000,1373  ;4723  D_K[.60]    ; Expected Interrupt vector
U 1373, 0001,003C,0180,09E8,0000,1074  ;4724  RC[0D]_D    ; Save for error logout
U 1074, 0000,003D,0180,0800,0000,1270  ;4725 =00 CALL[CHECK.VECTOR]  ; Was the correct vector generated ?
U 1075, 0000,003C,0180,0800,0000,1374  ;4726  J/VECTOK.7    ; OK.Correct vector was generated.
```

```
            ;4727  .PAGE
            ;4728 ;
            ;4729 ;////////////////////////////////////////////////////////////////////////////////
            ;4730 ;
U 1076, 0000,003D,D980,0800,0084,70B0  ;4731  ERROR1[.9]   ; Wrong vector was generated
            ;4732
            ;4733 ;
            ;4734 ;////////////////////////////////////////////////////////////////////////////////
            ;4735 ;
            ;4736 ; ERROR LOGOUT:
            ;4737 ;
            ;4738 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4739 ;  EXPECTED INTERRUPT VECTOR
            ;4740 ;  NOT USED
            ;4741 ;  NOT USED
            ;4742 ;  NOT USED
            ;4743 ;  NOT USED
            ;4744 ;  ID VECTOR REGISTER
            ;4745 ;  ID SBI ERROR REGISTER
            ;4746 ;  ID FAULT REGISTER
            ;4747 ;
            ;4748 ;
            ;4749 ;////////////////////////////////////////////////////////////////////////////////
            ;4750 ;
            ;4751 =
            ;4752 VECTOK.7:
U 1374, 0000,003C,6DF0,2C00,0000,1375  ;4753  Q_ID[FAULT]   ; Cpu Fault Set ?
U 1375, 000D,2034,0180,0A00,0010,1376  ;4754  ALU_Q[AND]R[0],CLK.UBCC  ; Test Cpu Fault Bit
U 1376, 0000,013C,0180,0800,0000,118E  ;4755  Z?
U 118E, 0000,003C,0180,0800,0000,1377  ;4756 =0 J/T7.S6.CFBOK   ; Ok Cpu Fault Set
```

```
            ;4757  .PAGE
            ;4758 ;
            ;4759 ;////////////////////////////////////////////////////////////////////////////////
            ;4760 ;
U 118F. 0000.003D.1980.0800.0084.70B0  ;4761  ERROR1[ZERO]   ; Cpu did not detect the timeout
            ;4762       ; ...Fault bit NOT set in ID Fault REG
            ;4763 ;
            ;4764 ;
            ;4765 ;////////////////////////////////////////////////////////////////////////////////
            ;4766 ;
            ;4767 ; ERROR LOGOUT:
            ;4768 ;
            ;4769 ; DATA: NO DATA LOGOUT
            ;4770 ;
            ;4771 ;////////////////////////////////////////////////////////////////////////////////
            ;4772 ;
            ;4773 T7.S6.CFBOK:
U 1377. 000D.2034.0180.0A40.0010.1378  ;4774  ALU_Q[AND]R[8],CLK.UBCC          ; Test SBI Fault Bit
U 1378. 0000.013C.0180.0800.0000.1190  ;4775  Z?     ; Check condition codes
U 119C. 0000.003C.0180.0800.0000.1379  ;4776 =0 J/T7.S6.SFBOK    ; Ok Bit was set
```

```
            ;4777  .PAGE
            ;4778 ;
            ;4779 ;////////////////////////////////////////////////////////////////////////////////
            ;4780 ;
U 1191. 0000.003D.1980.0800.0084.70B0  ;4781  ERROR1[ZERO]   ; SBI did NOT detect the timeout
            ;4782      ;...SBI Fault bit not set in Fault Reg
            ;4783 ;
            ;4784 ;////////////////////////////////////////////////////////////////////////////////
            ;4785 ;
            ;4786 ; ERROR LOGOUT:
            ;4787 ;
            ;4788 ; DATA: NO DATA LOGOUT
            ;4789 ;
            ;4790 ;////////////////////////////////////////////////////////////////////////////////
            ;4791 ;
            ;4792 T7.S6.SFBOK:
U 1379. 0011.2034.01C0.0948.0000.137A  ;4793  Q_ALU,ALU_Q[AND]RC[09]  ;MASK SBI STATUS BITS IN CPU
U 137A. 0001.203C.0180.09E8.0000.137B  ;4794  RC[0D]_Q    ;SAVE THEM
U 137B. 000D.2020.0180.0A50.0010.137C  ;4795  ALU_Q[XOR]R[0A],CLK.UBCC ;DID THE CPU RECORD ITSELF
            ;4796      ;  AS TRANSMITTER DURING FAULT ?
U 137C. 0800.003C.0180.0A50.0000.137D  ;4797  D_R[0A]     ; Set up expected bits for error logout
U 137D. 0001.013C.0180.09E0.0000.1192  ;4798  Z?,RC[0C]_D    : ...
            ;4799
```

```
            ;4800  .PAGE
            ;4801 ;
            ;4802 ;////////////////////////////////////////////////////////////////////////////////
            ;4803 ;
U 1192. 0000.003D.0D80.0800.0084.70B0  ;4804 =0 ERROR1[.3]   ; Cpu did not identify itself
            ;4805      ; ... as transmitter during fault
            ;4806 ;
            ;4807 ;
            ;4808 ;////////////////////////////////////////////////////////////////////////////////
            ;4809 ;
            ;4810 ; ERROR LOGOUT:
            ;4811 ;
            ;4812 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4813 ;  MASKED   SBI-FAULT-STATUS BITS READ (31-26)
            ;4814 ;  EXPECTED SBI-FAULT_STATUS BITS      (31-26)
            ;4815 ;
            ;4816 ;////////////////////////////////////////////////////////////////////////////////
            ;4817 ;
U 1193. 0010.0038.0180.0970.0200.137E  ;4818  VA_RC[0E]   ; Read Memory SBI Fault Status
U 137E. 0000.003C.0180.C800.0000.137F  ;4819  MCT/READ.P   ; ....
U 137F. 0001.003C.0180.09E0.0000.1380  ;4820  RC[0C]_D    ; Save Data Read
U 1380. 0818.0038.A580.0800.0000.1381  ;4821  D_K[.60]    ; Expected Interrupt vector
U 1381. 0001.003C.0180.09E8.0000.1078  ;4822  RC[0D]_D    ; Save for error logout
U 1078. 0000.003D.0180.0800.0000.1270  ;4823 =00 CALL[CHECK.VECTOR]  ; Was the correct vector generated ?
U 1079. 0000.003C.0180.0800.0000.1382  ;4824  J/VECTOK.8   ; OK. Correct vector was generated
```

```
            ;4825  .PAGE
            ;4826 ;
            ;4827 ;////////////////////////////////////////////////////////////////////////////////
            ;4828 ;
U 107A, 0000.003D.D980.0800.0084.70B0  ;4829  ERROR1[.9]   ; Wrong vector was generated
            ;4830
            ;4831 ;
            ;4832 ;////////////////////////////////////////////////////////////////////////////////
            ;4833 ;
            ;4834 ; ERROR LOGOUT:
            ;4835 ;
            ;4836 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4837 ;  EXPECTED INTERRUPT VECTOR
            ;4838 ;  NOT USED
            ;4839 ;  NOT USED
            ;4840 ;  NOT USED
            ;4841 ;  NOT USED
            ;4842 ;  ID VECTOR REGISTER
            ;4843 ;  ID SBI ERROR REGISTER
            ;4844 ;  ID FAULT REGISTER
            ;4845 ;
            ;4846 ;
            ;4847 ;////////////////////////////////////////////////////////////////////////////////
            ;4848 ;
            ;4849 =
            ;4850 VECTOK.8:
U 1382, 0810.0038.0180.0960.0000.1383  ;4851  D_RC[0C]   ; Put data read back in the D Register
U 1383, 0811.0034.0180.0948.0000.1384  ;4852  D_ALU,ALU_D[AND]rc[09]  ; Mask the SBI-Fault bits
U 1384, 0001.003C.0180.09E8.0000.1385  ;4853  RC[0D]_D   ; Save the Masked bits
U 1385, 0011.0020.0180.0918.0010.1386  ;4854  ALU_D[XOR]RC[3],CLK.UBCC ; Check for Interlock sequence fault
U 1386, 0010.0038.01C0.0918.0000.1387  ;4855  Q_RC[3]    ; Save the expected bits
U 1387, 0001.213C.0180.09E0.0000.1194  ;4856  Z?,RC[0C]_Q   ; Put the expected bits in RC[0C]
```

```
            ;4857   .PAGE
            ;4858  ;
            ;4859  ;////////////////////////////////////////////////////////////////////////////
            ;4860  ;
U 1194. 0000.003D.0D80.0800.0084.70B0   ;4861 =0 ERROR1[.3]    ; Wrong bits set in Cnfg Register A
            ;4862  ;
            ;4863  ;
            ;4864  ;////////////////////////////////////////////////////////////////////////////
            ;4865  ;
            ;4866  ; ERROR LOGOUT:
            ;4867  ;
            ;4868  ; DATA: I/O BASE ADDRESS OF MS780E CURRENTLY UNDER TEST
            ;4869  ;  MASKED   FAULT BITS FROM CNFG A
            ;4870  ;  EXPECTED FAULT BITS FROM CNFG A
            ;4871  ;
            ;4872  ;////////////////////////////////////////////////////////////////////////////
            ;4873  ;
U 1195. 0000.003C.0180.0800.0000.1196   ;4874   NOP
U 1196. 0000.003D.0180.0800.0000.10C0   ;4875 =0 CALL[SBI.INIT]    ; Clear the SBI Fault Register
U 1197. 0000.003C.0180.0800.0000.1388   ;4876   NOP     ; Clock ticks
U 1388. 0000.003C.0180.0800.0000.1389   ;4877   NOP
U 1389. 0000.003C.6DF0.2C00.0000.138A   ;4878   Q_ID[FAULT]    ; Check the SBI Fault Bits
U 138A. 000D.2034.0180.0A40.0010.138B   ;4879   ALU_Q[AND]R[8],CLK.UBCC   ; SBI Fault clear  ?
U 138B. 0000.013C.0180.0800.0000.1198   ;4880   Z?     ; Check condition codes
```

```
            ;4881   .PAGE
            ;4882 ;
            ;4883 ;////////////////////////////////////////////////////////////////////////////////
            ;4884 ;
U 1198, 0000,003D,1980,0800,0084,70B0  ;4885 =0 ERROR1[ZERO]   ; No SBI Fault bit set
            ;4886       ; ...should be clear
            ;4887 ;
            ;4888 ;////////////////////////////////////////////////////////////////////////////////
            ;4889 ;
            ;4890 ; ERROR LOGOUT:
            ;4891 ;
            ;4892 ; DATA: NO DATA LOGOUT
            ;4893 ;
            ;4894 ;////////////////////////////////////////////////////////////////////////////////
            ;4895 ;
U 1199, 000D,2034,0180,0A00,0010,138C  ;4896  ALU_Q[AND]R[0],CLK.UBCC  ; Cpu Fault clear ?
U 138C, 0000,013C,0180,0800,0000,119A  ;4897  Z?     ; Check condition codes
```

```
            ;4898  .PAGE
            ;4899 ;
            ;4900 ;////////////////////////////////////////////////////////////////////////////////
            ;4901 ;
U 119A, 0000,003D,1980,0800,0084,70B0  ;4902 =0 ERROR1[ZERO]   ; CPU Fault bit set should be clear
            ;4903 ;
            ;4904 ;
            ;4905 ;////////////////////////////////////////////////////////////////////////////////
            ;4906 ;
            ;4907 ; ERROR LOGOUT:
            ;4908 ;
            ;4909 ; DATA: NO DATA LOGOUT
            ;4910 ;
            ;4911 ;////////////////////////////////////////////////////////////////////////////////
            ;4912 ;
U 119B, 0011,2034,01C0,0948,0010,138D  ;4913  Q_ALU,ALU_Q[AND]rc[09],CLK.UBCC ; CPU SBI Fault status bits clear ?
U 138D, 0001,203C,0180,09E8,0000,138E  ;4914        RC[0D]_Q   ; Save Masked Bits for error logout
U 138E, 0003,013C,0180,09E0,0000,119C  ;4915  Z?,RC[0C]_0   ; Check condition codes
```

```
            ;4916  .PAGE
            ;4917  ;
            ;4918  ;////////////////////////////////////////////////////////////////////////////////
            ;4919  ;
U 119C, 0000,003D,0D80,0800,0084,70B0  ;4920 =0 ERROR1[.3]   ; NO CPU-Fault-Status bits not CLEAR
            ;4921  ;
            ;4922  ;
            ;4923  ;////////////////////////////////////////////////////////////////////////////////
            ;4924  ;
            ;4925  ; ERROR LOGOUT:
            ;4926  ;
            ;4927  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4928  ;  MASKED   SBI-FAULT-STATUS BITS READ
            ;4929  ;  EXPECTED SBI-FAULT STATUS BITS
            ;4930  ;
            ;4931  ;////////////////////////////////////////////////////////////////////////////////
            ;4932  ;
            ;4933  ;** SUBTEST 7 **
            ;4934  ;
            ;4935  ; INTERLOCK TIMING TESTS
            ;4936  ;
            ;4937  ; FIRST CHECK THE INTERLOCK READ,INTERLOCK READ SEQUENCE
            ;4938  ; THIS WILL ASSURE THE INTERLOCK TIME COUNTER IN
            ;4939  ; MEMORY CONTROLLER COUNT LESS THEN 512 TICKS(APPROX)
            ;4940  ; I.E. MEMORY COUNTER SHOULD FINISH BEFORE THE CPU TIMES
            ;4941  ; OUT
            ;4942  ;
            ;4943  ;////////////////////////////////////////////////////////////////////////////////
            ;4944  ;
U 119D, 0000,003C,0180,0800,0000,119E  ;4945  NOP
U 119E, 0000,003D,0180,0800,0000,11F0  ;4946 =0 SUBTEST    ; Increment the subtest number
U 119F, 0000,003C,0180,0800,0000,11A0  ;4947  NOP     ; ....
U 11A0, 0000,003D,0180,0800,0000,11E4  ;4948 =0 ERLOOP    ; Set U address for error loop
U 11A1, 0000,003C,0180,0800,0000,11A2  ;4949  NOP     ; ...
U 11A2, 0000,003D,0180,0800,0000,1283  ;4950 =0 CALL[INIT]   ; Initialize the SBI
U 11A3, 0000,003C,0180,0800,0000,11A4  ;4951  nop     ; ...
U 11A4, 0000,003D,8980,0800,0084,7231  ;4952 =0 SET.TRAP.MASK[.D]  ; Enable Time-Outs Utraps
U 11A5, 0010,0038,0180,0970,0200,138F  ;4953  VA_RC[0E]   ; Point CNFG A to VA
U 138F, 0000,003C,0180,E800,0000,107C  ;4954  MCT/LOCKREAD.P   ; Read CNFG A and Start the memory
            ;4955      ; ...Interlock Counter
U 107C, 0000,003D,0180,0800,0000,121A  ;4956 =00 CHECK.INTERRUPT   ; Check for Interrupts
U 107D, 0000,003C,0180,0800,0000,1080  ;4957  J/T7.S7.NOINT1   ; Jump if no Interrupts
```

```
          ;4958  .PAGE
          ;4959 ;
          ;4960 ;////////////////////////////////////////////////////////////////////////////////
          ;4961 ;
U 107E, 0000,003D,F580,0800,0084,70B0  ;4962  ERROR1[.A]   ; !!!UNEXPECTED INTERRUPT !!!!!
          ;4963      ; ...on first Interlock Read
          ;4964 ;
          ;4965 ;
          ;4966 ;////////////////////////////////////////////////////////////////////////////////
          ;4967 ;
          ;4968 ; ERROR LOGOUT:
          ;4969 ;
          ;4970 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
          ;4971 ;  NOT USED
          ;4972 ;  NOT USED
          ;4973 ;  NOT USED
          ;4974 ;  NOT USED
          ;4975 ;  NOT USED
          ;4976 ;  INTERRUPT VECTOR REGISTER
          ;4977 ;  SBI.ERROR REGISTER
          ;4978 ;  SBI.FAULT REGISTER
          ;4979 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
          ;4980 ;
          ;4981 ;////////////////////////////////////////////////////////////////////////////////
          ;4982 ;
          ;4983 =
          ;4984 =00
          ;4985 T7.S7.NOINT1:
U 1080, 0000,003D,0180,0800,0000,1227  ;4986  CHECK.UTRAP   ; See if Time-Out occured
U 1081, 0000,003C,0180,0800,0000,11A6  ;4987  J/T7.S7.NOTIMO1   ; Jump if no Time-Out
```

```
            ;4988  .PAGE
            ;4989 ;
            ;4990 ;////////////////////////////////////////////////////////////////////////////////
            ;4991 ;
U 1082. 0000.003D.0180.0800.0084.70B0  ;4992  ERROR1[.8]   ; Time-Out on first interrupt read
            ;4993 ;
            ;4994 ;
            ;4995 ;////////////////////////////////////////////////////////////////////////////////
            ;4996 ;
            ;4997 ; ERROR LOGOUT:
            ;4998 ;
            ;4999 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5000 ;  NOT USED
            ;5001 ;  NOT USED
            ;5002 ;  NOT USED
            ;5003 ;  NOT USED
            ;5004 ;  NOT USED
            ;5005 ;  EXPECTED UTRAP FLAG
            ;5006 ;  RECIEVED UTRAP FLAG
            ;5007 ;
            ;5008 ;////////////////////////////////////////////////////////////////////////////////
            ;5009 ;
            ;5010 =
            ;5011 =0
            ;5012 T7.S7.NOTIMO1:
U 11A6. 0000.003D.8980.0800.0084.7231  ;5013  SET.TRAP.MASK[.D]  ; Enable Time Outs
U 11A7. 0000.003C.0180.E800.0000.1084  ;5014  MCT/LOCKREAD.P    ; Start the second Interlock READ
U 1084. 0000.003D.0180.0800.0000.121A  ;5015 =00 CHECK.INTERRUPT    ; Check for Interrupts
U 1085. 0000.003C.0180.0800.0000.1088  ;5016  J/T7.S7.NOINT2    ; Jump if no Interrupts
```

```
ZZ-ESKAR-V22  TEST 18D SBI FORCED FAULT TESTS                                   SEQ 0340
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23          Page  15

            ;5017  .PAGE
            ;5018 ;
            ;5019 ;////////////////////////////////////////////////////////////////////////////
            ;5020 ;
U 1086, 0000,003D,F580,0800,0084,70B0  ;5021  ERROR1[.A]   ; !!!UNEXPECTED INTERRUPT !!!!!
            ;5022       ; ...on second Interlock Read
            ;5023 ;
            ;5024 ;
            ;5025 ;////////////////////////////////////////////////////////////////////////////
            ;5026 ;
            ;5027 ; ERROR LOGOUT:
            ;5028 ;
            ;5029 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5030 ;  NOT USED
            ;5031 ;  NOT USED
            ;5032 ;  NOT USED
            ;5033 ;  NOT USED
            ;5034 ;  NOT USED
            ;5035 ;  INTERRUPT VECTOR REGISTER
            ;5036 ;  SBI.ERROR REGISTER
            ;5037 ;  SBI.FAULT REGISTER
            ;5038 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;5039 ;
            ;5040 ;////////////////////////////////////////////////////////////////////////////
            ;5041 ;
            ;5042 =
            ;5043 =00
            ;5044 T7.S7.NOINT2:
U 1088, 0000,003D,0180,0800,0000,1227  ;5045  CHECK.UTRAP   ; See if Time-Out occured
U 1089, 0000,003C,0180,0800,0000,11A8  ;5046  J/T7.S7.NOTIM02   ; Jump if no Time-Out
```

```
            ;5047   .PAGE
            ;5048 ;
            ;5049 ;////////////////////////////////////////////////////////////////////////////////
            ;5050 ;
U 108A, 0000,003D,0180,0800,0084,70B0  ;5051  ERROR1[.8]   ; Time-Out on second interrupt read
            ;5052 ;
            ;5053 ;
            ;5054 ;////////////////////////////////////////////////////////////////////////////////
            ;5055 ;
            ;5056 ; ERROR LOGOUT:
            ;5057 ;
            ;5058 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5059 ;  NOT USED
            ;5060 ;  NOT USED
            ;5061 ;  NOT USED
            ;5062 ;  NOT USED
            ;5063 ;  NOT USED
            ;5064 ;  EXPECTED UTRAP FLAG
            ;5065 ;  RECIEVED UTRAP FLAG
            ;5066 ;
            ;5067 ;////////////////////////////////////////////////////////////////////////////////
            ;5068 ;
            ;5069 ;** SUBTEST 8 **
            ;5070 ;
            ;5071 ; CHECK TO SEE COUNTER IN MEMORY COUNTS APPROX 500
            ;5072 ; BUS CYCLES
            ;5073 ;
            ;5074 ; INITIATE AN INTERLOCK READ , STARTING THE
            ;5075 ; COUNTER. WAIT APPROX 500 CLOCKS TICS.
            ;5076 ; INITIATE AN INTERLOCK WRITE, SEEING NO INTERLOCK
            ;5077 ; SEQUENCE FAULT.
            ;5078 ;
            ;5079 ;////////////////////////////////////////////////////////////////////////////////
            ;5080 ;
            ;5081 =
            ;5082 =0
            ;5083 T7.S7.NOTIMO2:
U 11A8, 0000,003D,0180,0800,0000,11F0  ;5084  SUBTEST    ; Increment the subtest number
U 11A9, 0000,003C,0180,0800,0000,11AA  ;5085  NOP     ; ...
U 11AA, 0000,003D,0180,0800,0000,11E4  ;5086 =0 ERLOOP     ; Set Uaddress for error loop
U 11AB, 0000,003C,0180,0800,0000,11AC  ;5087  NOP     ; ...
U 11AC, 0000,003D,0180,0800,0000,1283  ;5088 =0 CALL[INIT]    ; Initialize the SBI
U 11AD, 0F00,003C,0180,0800,0000,1390  ;5089  D_0     ; Point to location 0
U 1390, 0001,003C,0180,0800,0200,1391  ;5090  VA_D     ; ...In the VA register
U 1391, 0000,003C,0180,E800,0000,1392  ;5091  MCT/LOCKREAD.P    ; Start an Interlock Read
U 1392, 0818,0038,C580,0800,0000,1393  ;5092  D_K[.88]    ; Get number of cycles to wait
U 1393, 0819,0014,1180,0800,0000,11AE  ;5093  D_D+K[.4]    ; ....
U 11AE, 0000,003D,0180,0800,0000,1219  ;5094 =0 CALL[WAIT.LOOP]    ; Start a wait loop of 500 cycles
U 11AF, 0000,003C,0180,0800,0000,11B0  ;5095  NOP     ; ...
U 11B0, 0000,003D,8980,0800,0084,7231  ;5096 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out Traps
U 11B1, 0F00,003C,0180,0800,0000,1394  ;5097  D_0     ; Get data to write to memory
U 1394, 0000,003C,0180,B800,0000,11B2  ;5098  MCT/LOCKWRITE.P    ; Execute an Interlock write
            ;5099 =0 D_K[.FF],    ; Get number of cycle to wait
U 11B2, 0818,0039,4980,0800,0000,1219  ;5100   CALL[WAIT.LOOP]  ; Allow SBI to finish write cycle
U 11B3, 0000,003C,0180,0800,0000,108C  ;5101  NOP     ; ...
```

```
U 108C, 0000,003D,0180,0800,0000,121A  ;5102 =00 CHECK.INTERRUPT    ; Any unexpected interrupts ?
U 108D, 0000,003C,0180,0800,0000,1090  ;5103  J/T7.SB.NOINT1    ; Jump if no interrupts
```

```
            ;5104  .PAGE
            ;5105 ;
            ;5106 ;////////////////////////////////////////////////////////////////////////////////
            ;5107 ;
U 108E, 0000,003D,F580,0800,0084,70B0  ;5108  ERROR1[.A]   ; Unexpected Interrupt on Interlock
            ;5109       ; ...Write with 500 cycles delay
            ;5110       ; ...from the Interlock Read
            ;5111 ;
            ;5112 ;
            ;5113 ;////////////////////////////////////////////////////////////////////////////////
            ;5114 ;
            ;5115 ; ERROR LOGOUT:
            ;5116 ;
            ;5117 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5118 ;  NOT USED
            ;5119 ;  NOT USED
            ;5120 ;  NOT USED
            ;5121 ;  NOT USED
            ;5122 ;  NOT USED
            ;5123 ;  INTERRUPT VECTOR REGISTER
            ;5124 ;  SBI.ERROR REGISTER
            ;5125 ;  SBI.FAULT REGISTER
            ;5126 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALAS=0)
            ;5127 ;
            ;5128 ;////////////////////////////////////////////////////////////////////////////////
            ;5129 ;
U 108F, 0000,003C,0180,0800,0000,1090  ;5130  NOP     ;
            ;5131 =00
            ;5132 T7.58.NOINT1:
U 1090, 0000,003D,0180,0800,0000,1227  ;5133  CHECK.UTRAP    ; Any Time-Outs utraps occur ?
U 1091, 0000,003C,0180,0800,0000,1393  ;5134  J/T7.58.NOTIMO1    ; Jump if no time-outs
```

```
            ;5135  .PAGE
            ;5136 ;
            ;5137 ;////////////////////////////////////////////////////////////////////////////////
            ;5138 ;
U 1092, 0000,003D,0180,0800,0084,70B0  ;5139  ERROR1[.8]   ; Interlock Write Timeout
            ;5140 ;
            ;5141 ;
            ;5142 ;////////////////////////////////////////////////////////////////////////////////
            ;5143 ;
            ;5144 ; ERROR LOGOUT:
            ;5145 ;
            ;5146 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5147 ;  NOT USED
            ;5148 ;  NOT USED
            ;5149 ;  NOT USED
            ;5150 ;  NOT USED
            ;5151 ;  NOT USED
            ;5152 ;  EXPECTED UTRAP FLAG
            ;5153 ;  RECIEVED UTRAP FLAG
            ;5154 ;
            ;5155 ;////////////////////////////////////////////////////////////////////////////////
            ;5156 ;
U 1093, 0000,003C,0180,0800,0000,1395  ;5157  NOP
            ;5158 T7.S8.NOTIMO1:
U 1395, 0000,003C,0180,0800,0000,1030  ;5159  J/RESTART.TEST.7
            ;5160 TEST.7.EXIT:
U 1396, 0000,003C,0180,0800,0000,11B4  ;5161  NOP   ; TEST 7 FINISHED
            ;5162
```

```
;5163  .PAGE "TEST 18E CPU AND MEMORY MULT XMIT FLT"
;5164 ;*************************************************************
;5165 ;+++
;5166 ;
;5167 ; Functional Description:
;5168 ; ---------------------
;5169 ;
;5170 ;      This test keeps  a  running  count  in  the  Maintenance  ID
;5171 ; Register  (ID  reg.   1D),  while  forcing a Multiple Transmitter
;5172 ; Fault.  A check is made to see that the CPU and Memory,  each  in
;5173 ; turn,  will  detect the fault when putting Write Data on the SBI.
;5174 ; This tests the ID Comparator logic on the M8219 SBH  module,  and
;5175 ; the Multiple Transmitter bit in the SBI Interface CNFG A register
;5176 ; (bit 27).  Also, a check will be made to see that bit 26 in  CNFG
;5177 ; A register (Memory was TRANSMITTER DURING FAULT) is asserted.
;5178 ;
;5179 ;
;5180 ; SUBTEST 1 - CPU DETECTION OF MULTIPLE XMITTER FAULT -
;5181 ;
;5182 ;      This Subtest keeps a count  pattern  through  the  ID  MAINT
;5183 ; Register    (<27:23>),   while   forcing   multiple   transmitter
;5184 ; faults.  It checks that the SBI Fault Register bit  27  gets  set
;5185 ; when  a multiple transmitter fault occurs for any other ID except
;5186 ; the CPU ID.
;5187 ;
;5188 ;
;5189 ; SUBTEST 2 - MEMORY DETECTION OF MULTIPLE XMITTER FAULT -
;5190 ;
;5191 ;      Subtest 2 is very much similar to  Subtest  1,  except  that
;5192 ; this  time CNF Reg A of the MS780-E under test is checked to make
;5193 ; sure that the Multiple Transmitter bit is set for all ID's except
;5194 ; the CPU's ID.
;5195 ;
;5196 ;
;5197 ; Logic Description:
;5198 ; -----------------
;5199 ;
;5200 ;   N/A
;5201 ;
;5202 ; Error Logout:
;5203 ; ------------
;5204 ;
;5205 ; See individual error reports.
;5206 ;
;5207 ; Sync Point Description:
;5208 ; ----------------------
;5209 ;
;5210 ;   N/A
;5211 ;
;5212 ;=============================================================
;5213 ;
;5214 ; Register Map:
;5215 ; ------------
;5216 ;
;5217 ;     RA,RB     Description:
```

```
                ;5218 ;      -----      -----------
                ;5219 ;
                ;5220 ; 4 x00018000 ( BITS TO TURN OFF CACHE IN ID MAINT REG)
                ;5221 ;
                ;5222 ; 5 X10000000 ( BIT 28 IN ID MAINT )
                ;5223 ;
                ;5224 ;       RC       Description:
                ;5225 ;       --       -----------
                ;5226 ;
                ;5227 ; 8 ID COUNTER
                ;5228 ;
                ;5229 ;        C CPU ID
                ;5230 ;
                ;5231 ;        E       I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
                ;5232 ;
                ;5233 ;
                ;5234 ;--
                ;5235 ;*************************************************************
                ;5236 ;
                ;5237 =0
U 11B4. 0000.003D.0180.0800.0000.1063   ;5238 TEST.8: NEWTST    ; Increment test number
U 11B5. 0000.003C.0180.0800.0000.1094   ;5239  NOP
U 1094. 0000.003D.0180.0800.0000.10C8   ;5240 =00 CALL[FIND.MS780E]  ; Find MS780-E/H on the SBI
U 1095. 0000.003C.0180.0800.0000.13DD   ;5241  J/TEST.8.EXIT   ; No MS780-E/Hs found
U 1096. 0000.003C.0180.0800.0000.1098   ;5242  NOP
                ;5243 =
                ;5244 =00
                ;5245 RESTART.TEST.8:    ; Entry point for the next MS780-E/H
                ;5246  D_K[.4],    ; Prepare to select Internal Interleave
U 1098. 0818.0039.1180.0800.0000.11F2   ;5247   CALL[SELECT.CNTRLR] ; ...Mode
U 1099. 0000.003C.0180.0800.0000.10A0   ;5248  J/TEST.8.IMIS   ; Internal Interleave Misconfiguration
                ;5249 =0 RC[0D]_K[.1],    ; Internal Interleave is OK.
U 109A. 0018.0039.0580.09E8.0000.11B6   ;5250   CALL[TEST.8.TST] ; ...execute the test
U 109B. 0000.003C.0180.0800.0000.109C   ;5251  NOP
U 109C. 0000.003D.0180.0800.0000.1275   ;5252 =00 CALL[ENABLE.NEXT.MS780E] ; Go find next Ms780e
U 109D. 0000.003C.0180.0800.0000.1098   ;5253  J/RESTART.TEST.8
U 109E. 0000.003C.0180.0800.0000.13DD   ;5254  J/TEST.8.EXIT   ; Exit when done with the test
                ;5255 =
                ;5256 =00
                ;5257 TEST.8.IMIS:
U 10A0. 0000.003D.0180.0800.0000.1028   ;5258  CALL[SELECT.UPPER]  ; Try now to enable the Upper
                ;5259       ; ...Controller
U 10A1. 0000.003C.0180.0800.0000.1397   ;5260  J/TEST.8.UMIS   ; Upper Controller is Misconfigured
U 10A2. 0000.003D.0180.0800.0000.11B6   ;5261 =0 CALL[TEST.8.TST]  ; Upper Controller has been enabled
                ;5262       ; ...go execute the test
U 10A3. 0000.003C.0180.0800.0000.10A4   ;5263  NOP
U 10A4. 0000.003D.0180.0800.0000.1275   ;5264 =00 CALL[ENABLE.NEXT.MS780E] ; Go find next Ms780e
U 10A5. 0000.003C.0180.0800.0000.1098   ;5265  J/RESTART.TEST.8
U 10A6. 0000.003C.0180.0800.0000.13DD   ;5266  J/TEST.8.EXIT   ; Exit when done with the test
                ;5267 =
                ;5268 TEST.8.UMIS:
U 1397. 0018.0038.0D80.09E8.0000.10A8   ;5269  RC[0D]_K[.3]   ; Set flag for the fail-chain
U 10A8. 0000.003D.0180.0800.0000.1024   ;5270 =00 CALL[SELECT.LOWER]  ; Try now to enable the Lower
                ;5271       ; ...Controller
```

```
            ;5272  .PAGE
            ;5273 ;
            ;5274 ;////////////////////////////////////////////////////////////////////////////////
            ;5275 ;
U 10A9, 0000.003D.0980.0800.0084.70B0  ;5276  ERROR1[.2]   ; Both Controllers are Misconfigured
            ;5277       ; ...
            ;5278 ;
            ;5279 ;////////////////////////////////////////////////////////////////////////////////
            ;5280 ;
            ;5281 ; ERROR LOGOUT:
            ;5282 ;
            ;5283 ; DATA:
            ;5284 ;  I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5285 ;  ERROR CODE ( CONTROLLERS MISCONFIGURED )
            ;5286 ;
            ;5287 ;
            ;5288 ;////////////////////////////////////////////////////////////////////////////////
            ;5289 ;
U 10AA, 0000.003D.01B0.0800.0000.11B6  ;5290 =0 CALL[TEST.8.TST]  ; Lower Controller is enabled
U 10AB, 0000.003C.01B0.0800.0000.10AC  ;5291  NOP
U 10AC, 0000.003D.01B0.0800.0000.1275  ;5292 =00 CALL[ENABLE.NEXT.MS780E] ; Go find next Ms780e
U 10AD, 0000.003C.01B0.0800.0000.1098  ;5293  J/RESTART.TEST.8
U 10AE, 0000.003C.01B0.0800.0000.13DD  ;5294  J/TEST.8.EXIT   ; Exit when test is done
            ;5295 =
            ;5296 ;
            ;5297 ;********************************************************************************
            ;5298 ;////////////////////////////////////////////////////////////////////////////////
            ;5299 ;**
            ;5300 ;
            ;5301 ; THIS IS THE MAIN BODY OF THE TEST WHERE THE CONDITIONS
            ;5302 ; ARE BEING CHECKED OUT.
            ;5303 ;
            ;5304 ;--
            ;5305 ;////////////////////////////////////////////////////////////////////////////////
            ;5306 ;
            ;5307 =0
            ;5308 TEST.8.TST:
U 11B6, 0000.003D.0180.0800.0000.10C0  ;5309  CALL[SBI.INIT]   ; Make sure the SBI is clean
U 11B7, 0000.003C.0180.0800.0000.11B8  ;5310  NOP
U 11B8, 0000.003D.0180.0800.0000.1266  ;5311 =0 CALL[FIND.CPU.ID]  ; Find the CPU ID
U 11B9, 0001.003C.0180.09E0.0000.1398  ;5312  RC[0C]_D   ; Save CPU ID in RC[0C]
U 1398, 0818.0038.7DF8.0800.0000.1399  ;5313  D_K[.18].Q_0   ; Form x00018000
U 1399, 0000.003C.8580.0800.0084.739A  ;5314  SC_K[.C]   ; ...
U 139A, 0D00.003C.0180.0800.0000.139B  ;5315  D_DAL.SC   ; ...D = x00018000
            ;5316  R[4]_D,    ; Save in R[4]
U 139B, 0F01.003C.0180.0AA0.0000.139C  ;5317   D_0   ; Clear the D register
U 139C, 0019.0010.ED80.0800.0082.139D  ;5318  SC_ALU,ALU_D+K[.1B]+1   ; SC = x1C
U 139D, 0003.001C.0180.0AA8.0000.139E  ;5319  R[5]_NOT.MASK   ; R[5] = x10000000
U 139E, 0818.0038.9D80.0800.0000.139F  ;5320  D_K[.7C]   ; Form x7C000000
U 139F, 0B00.003C.0180.0800.0000.13A0  ;5321  D_D.SWAP   ; D = x7C000000
U 13A0, 0001.003C.0180.0AB0.0000.13A1  ;5322  R[6]_D    ; Save in R[6]
U 13A1, 0818.0038.8580.0800.0000.13A2  ;5323  D_K[.C]    ; Form x0C000000
U 13A2, 0B00.003C.0180.0800.0000.13A3  ;5324  D_D.SWAP   ; D = x0C000000
U 13A3, 0001.003C.0180.0AB8.0000.11BA  ;5325  R[7]_D    ; Save in R[7]
            ;5326 ;
```

```
            ;5327 ;////////////////////////////////////////////////////////////////////////////////
            ;5328 ;**
            ;5329 ;
            ;5330 ; SUBTEST 1
            ;5331 ;
            ;5332 ;--
            ;5333 ;////////////////////////////////////////////////////////////////////////////////
            ;5334 ;
U 11BA, 0000,003D,0180,0800,0000,11F0  ;5335 =0 SUBTEST    ; Increment subtest number
U 11BB, 0003,003C,0180,09C0,0000,11BC  ;5336  RC[8]_0     ; Initialize ID Counter
            ;5337 =0
            ;5338 TEST.8.LOOP1:
U 11BC, 0000,003D,0180,0800,0000,11E4  ;5339  ERLOOP      ; Set up the Error looping mechanism
U 11BD, 0000,003C,0180,0800,0000,11BE  ;5340  NOP
U 11BE, 0000,003D,0180,0800,0000,10C0  ;5341 =0 CALL[SBI.INIT]   ;
U 11BF, 0000,003C,0180,0800,0000,13A4  ;5342  NOP
U 13A4, 081D,0038,0180,0940,0000,13A5  ;5343  D_RC[8]      ; Get ID Counter
U 13A5, 0000,003C,7D80,0800,0084,73A6  ;5344  SC_K[.18]    ; Prepare to get x17 in SC
            ;5345  SC_SC-K[.1],    ; ...SC = x17
U 13A6, 0000,003C,05F8,0800,0084,B3A7  ;5346   Q_0    ; Clear the Q Register
U 13A7, 0D00,003C,0180,0800,0000,13A8  ;5347  D_DAL.SC    ; Shift the ID Counter bits into
            ;5348       ; ...position to be loaded in ID MAINT
U 13A8, 081C,2030,0180,0A20,0000,13A9  ;5349  D_D.OR.R[4]    ; Disable Cache
U 13A9, 081C,2030,0180,0A28,0000,13AA  ;5350  D_D.OR.R[5]    ; Set "Force Mult Xmit Flt" bit
U 13AA, 0000,003C,0180,0800,0000,13AB  ;5351  NOP
U 13AB, 0000,003C,7580,3C00,0000,13AC  ;5352  ID[MAINT]_D    ; Load the ID Maint Register
            ;5353  VA_K[ZERO],    ; Point to memory location 0
U 13AC, 0F18,0038,1980,0800,0200,11C0  ;5354   D_0    ; Get data to write to memory
U 11C0, 0000,003D,8980,0800,0084,7231  ;5355 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 11C1, 0000,003C,0180,A800,0000,13AD  ;5356  CACHE.P_D[LONG]    ; Force the Multiple Xmit Fault
U 13AD, 0800,003C,0180,0A20,0000,13AE  ;5357  D_R[4]     ; Prepare to clear bit 28 in Maint reg
U 13AE, 0000,003C,7580,3C00,0000,13AF  ;5358  ID[MAINT]_D    ; ...clear the bit
U 13AF, 0000,003C,6DF0,2C00,0000,13B0  ;5359  Q_ID[FAULT]    ; Get  ID Fault Register
U 13B0, 0001,203C,0180,09C8,0000,13B1  ;5360  RC[9]_Q     ; Save for error logout
U 13B1, 0800,003C,0180,0A30,0000,13B2  ;5361  D_R[6]     ; Get mask
U 13B2, 001D,0034,0180,09D0,0000,13B3  ;5362  RC[0A]_D.AND.Q    ; Mask unwanted bits
U 13B3, 0810,0038,0180,0960,0000,13B4  ;5363  D_RC[0C]    ; Get CPU ID
            ;5364  ALU_D.XOR.RC[8],   ; Is CPU ID equal to the ID Counter?
U 13B4, 0011,0020,0180,0940,0010,13B5  ;5365   CLK.UBCC  ; ...
U 13B5, 0000,013C,0180,0800,0000,11C2  ;5366  Z?     ; ...
U 11C2, 0000,003C,0180,0800,0000,13B8  ;5367 =0 J/TEST.8.NOCPID    ; ID Counter is Not equal to CPU ID
            ;5368       ; ID Counter is equal to CPU ID
U 11C3, 0003,003C,0180,09D8,0000,13B6  ;5369  RC[0B]_0    ; ...thus set the expected status of
            ;5370       ; ...bits in ID Fault Register
            ;5371  ALU_RC[0A],    ; See if the returned bit values
U 13B6, 0010,0038,0180,0950,0010,13B7  ;5372   CLK.UBCC  ; ...are equal to the expected.
U 13B7, 0000,013C,0180,0800,0000,11C4  ;5373  Z?     ;
```

```
            ;5374  .PAGE
            ;5375 ;
            ;5376 ;////////////////////////////////////////////////////////////////////////////////
            ;5377 ;
U 11C4, 0000,003D,5D80,0800,0084,70B2  ;5378 =0 ERROR2[.7]   ; Some bits are set in ID FAULT
            ;5379       ; ...Register
            ;5380 ;
            ;5381 ;////////////////////////////////////////////////////////////////////////////////
            ;5382 ;
            ;5383 ; ERROR LOGOUT:
            ;5384 ;
            ;5385 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5386 ;  NOT USED
            ;5387 ;  CPU ID
            ;5388 ;  EXPECTED STATUS OF THE ERROR BITS
            ;5389 ;  RECEIVED STATUS OF THE ERROR BITS
            ;5390 ;  RECEIVED CONTENTS OF ID FAULT REGISTER
            ;5391 ;  ID COUNTER
            ;5392 ;
            ;5393 ;
            ;5394 ;////////////////////////////////////////////////////////////////////////////////
            ;5395 ;
U 11C5, 0000,003C,0180,0800,0000,11C7  ;5396  J/TEST.8.NXTID1   ; Bits in ID FAULT are OK. Go and
            ;5397       ; ...get next ID
            ;5398 TEST.8.NOCPID:
U 13B8, 0800,003C,0180,0A38,0000,13B9  ;5399  D_R[7]    ; Get expected status of the fault bits
U 13B9, 0001,003C,0180,09D0,0000,13BA  ;5400  RC[0A]_D   ; Save for error logout
            ;5401  ALU_D.XOR.RC[0A],  ; See if the Received Fault bit status
U 13BA, 0011,0020,0180,0950,0010,13BB  ;5402   CLK.UBCC  ; ...is equal to the expected
U 13BB, 0000,013C,0180,0800,0000,11C6  ;5403  Z?    ; ...
```

```
            ;5404  .PAGE
            ;5405 ;
            ;5406 ;////////////////////////////////////////////////////////////////////////////////
            ;5407 ;
U 11C6, 0000,003D,5D80,0800,0084,70B2  ;5408 =0 ERROR2[.7]   ; Wrong Fault bits latched in ID
            ;5409       ; ...Fault Register
            ;5410 ;
            ;5411 ;////////////////////////////////////////////////////////////////////////////////
            ;5412 ;
            ;5413 ; ERROR LOGOUT:
            ;5414 ;
            ;5415 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5416 ;  NOT USED
            ;5417 ;  CPU ID
            ;5418 ;  EXPECTED STATUS OF THE ERROR BITS
            ;5419 ;  RECEIVED STATUS OF THE ERROR BITS
            ;5420 ;  RECEIVED CONTENTS OF ID FAULT REGISTER
            ;5421 ;  ID COUNTER
            ;5422 ;
            ;5423 ;
            ;5424 ;////////////////////////////////////////////////////////////////////////////////
            ;5425 ;
            ;5426 TEST.8.NXTID1:
U 11C7, 0810,0038,0180,0940,0000,13BC  ;5427  D_RC[8]    ; Get ID Counter
            ;5428  ALU_D[XOR]K[.20],   ; Is the ID equal to x20?
U 13BC, 0019,0020,7580,0800,0010,13BD  ;5429   CLK.UBCC  ; ...
U 13BD, 0000,013C,0180,0800,0000,11C8  ;5430  Z?    ; ...
            ;5431 =0 RC[8]_D+K[.1],   ; ID Counter not x20 yet.
U 11C8, 0019,0014,0580,09C0,0000,11BC  ;5432   J/TEST.8.LOOP1  ; Repeat the loop
U 11C9, 0000,003C,0180,0800,0000,11CA  ;5433  NOP
            ;5434 ;
            ;5435 ;////////////////////////////////////////////////////////////////////////////////
            ;5436 ;**
            ;5437 ;
            ;5438 ; SUBTEST 2
            ;5439 ;
            ;5440 ;--
            ;5441 ;////////////////////////////////////////////////////////////////////////////////
U 11CA, 0000,003D,0180,0800,0000,11F0  ;5442 =0 SUBTEST    ; Increment subtest number
U 11CB, 0818,0038,0580,0800,0000,13BE  ;5443  D_K[.1]    ; Initialize ID Counter
U 13BE, 0001,003C,0180,09C0,0000,11CC  ;5444  RC[8]_D    ; ...to 1
            ;5445 =0
            ;5446 TEST.8.LOOP2:
U 11CC, 0000,003D,0180,0800,0000,11E4  ;5447  ERLOOP    ; Set up the Error looping mechanism
U 11CD, 0000,003C,0180,0800,0000,11CE  ;5448  NOP
U 11CE, 0000,003D,0180,0800,0000,10C0  ;5449 =0 CALL[SBI.INIT]   ;
U 11CF, 0F18,0038,1980,0800,0200,13BF  ;5450  VA_K[ZERO].D_0   ; Point to location 0
U 13BF, 0000,003C,0180,A800,0000,13C0  ;5451  CACHE.P_D[LONG]   ; Inititalize location 0 to 0
U 13C0, 0818,0038,7D80,0800,0000,13C1  ;5452  D_K[.18]   ; Prepare to get x17 in SC
U 13C1, 0019,0000,0580,0800,0082,13C2  ;5453  SC_D-K[.1]   ; ...SC = x17
            ;5454  D_RC[8],   ; Get ID Counter
U 13C2, 0810,0038,01F8,0940,0000,13C3  ;5455   Q_0   ; Clear the Q Register
U 13C3, 0D00,003C,0180,0800,0000,13C4  ;5456  D_DAL.SC   ; Shift the ID Counter bits into
            ;5457       ; ...position to be loaded in ID MAINT
U 13C4, 081C,2030,0180,0A20,0000,13C5  ;5458  D_D.OR.R[4]   ; Disable Cache
```

```
U 13C5, 081C,2030,0180,0A28,0000,13C6  ;5459  D_D.OR.R[5]    ; Set "Force Mult Xmit Flt" bit
U 13C6, 0000,003C,7580,3C00,0000,13C7  ;5460  ID[MAINT]_D    ; Load the ID Maint Register
U 13C7, 0018,0038,1980,0800,0200,11D0  ;5461  VA_K[ZERO]    ; Point to memory location 0
U 11D0, 0000,003D,8980,0800,0084,7231  ;5462 =0 SET.TRAP.MASK[.D]  ; Enable Time-Out uTraps
U 11D1, 0000,003C,0180,C800,0000,13C8  ;5463  D[LONG]_CACHE.P    ; Force the Multiple Xmit Fault
U 13C8, 0800,003C,0180,0A20,0000,13C9  ;5464  D_R[4]     ; Prepare to clear bit 28 in Maint reg
U 13C9, 0000,003C,7580,3C00,0000,13CA  ;5465  ID[MAINT]_D    ; ...clear the bit
U 13CA, 0010,0038,0180,0970,0200,13CB  ;5466  VA_RC[0E]    ; Point to CNFG Reg A of the memory
U 13CB, 0000,003C,0180,C800,0000,13CC  ;5467  D[LONG]_CACHE.P    ; Read contents of CNFG Reg A
U 13CC, 0001,003C,0180,09C8,0000,13CD  ;5468  RC[9]_D     ; Save for error logout in RC[9]
U 13CD, 0000,003C,01C0,0A30,0000,13CE  ;5469  Q_R[6]     ; Get mask
U 13CE, 001D,0034,0180,09D0,0000,13CF  ;5470  RC[0A]_D.AND.Q    ; Mask unwanted bits and save
U 13CF, 0810,0038,0180,0960,0000,13D0  ;5471  D_RC[0C]    ; Get CPU ID
          ;5472  ALU_D.XOR.RC[8],  ; Is ID Counter equal to CPU ID?
U 13D0, 0011,0020,0180,0940,0010,13D1  ;5473   CLK.UBCC  ; ...
U 13D1, 0000,013C,0180,0800,0000,11D2  ;5474  Z?     ; ...
U 11D2, 0000,003C,0180,0800,0000,13D7  ;5475 =0 J/TEST.8.NCPID2   ; ID Counter is not yet equal to CPU ID
          ;5476       ; ID Counter is equal to CPU ID thus
U 11D3, 0003,003C,0180,09D8,0000,13D2  ;5477  RC[0B]_0    ; ...set the expected status of the
          ;5478       ; ...error bits in CNFG Reg A
U 13D2, 0818,0038,0180,0800,0000,13D3  ;5479  D_K[.8]     ; Get mask for MX bit in CNFG Reg A
U 13D3, 0B00,003C,0180,0800,0000,13D4  ;5480  D_D.SWAP    ; ....
U 13D4, 0010,0038,01C0,0950,0000,13D5  ;5481  Q_RC[0A]    ; Get received contents of CNFG Reg A
          ;5482  RC[0A]_D.AND.Q,   ; Check ONLY MX bit in CNFG Reg A
U 13D5, 001D,0034,0180,09D0,0010,13D6  ;5483   CLK.UBCC  ; ...and see that it is NOT set
U 13D6, 0000,013C,0180,0800,0000,11D4  ;5484  Z?     ; ...
```

```
            ;5485  .PAGE
            ;5486 ;
            ;5487 ;////////////////////////////////////////////////////////////////////////////////
            ;5488 ;
U 11D4, 0000,003D,5D80,0800,0084,70B2  ;5489 =0 ERROR2[.7]   ; Memory logged a MX error for
            ;5490       ; ...CPU ID
            ;5491 ;
            ;5492 ;////////////////////////////////////////////////////////////////////////////////
            ;5493 ;
            ;5494 ; ERROR LOGOUT:
            ;5495 ;
            ;5496 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5497 ;  NOT USED
            ;5498 ;  CPU ID
            ;5499 ;  EXPECTED STATUS OF ERROR BITS FORM CNFG REG A
            ;5500 ;  RECEIVED STATUS OF ERROR BITS FROM CNFG REG A
            ;5501 ;  RECEIVED CONTENTS OF THE REGISTER
            ;5502 ;  ID COUNTER
            ;5503 ;
            ;5504 ;
            ;5505 ;////////////////////////////////////////////////////////////////////////////////
            ;5506 ;
U 11D5, 0000,003C,0180,0800,0000,11D7  ;5507  J/TEST.8.NXTID2   ; Go increment ID Counter
            ;5508 TEST.8.NCPID2:
U 13D7, 0800,003C,0180,0A38,0000,13D8  ;5509  D_R[7]    ; Expected status of bit 27 and
            ;5510       ; ...26 in CNFG Reg A
U 13D8, 0001,003C,0180,09D8,0000,13D9  ;5511  RC[0B]_D   ; Save for error logout
            ;5512  ALU_D.XOR.RC[0A],  ; Check to see if the bits are set
U 13D9, 0011,0020,0180,0950,0010,13DA  ;5513   CLK.UBCC  ; ...
U 13DA, 0000,013C,0180,0800,0000,11D6  ;5514  Z?
            ;5515 ;
            ;5516 ;////////////////////////////////////////////////////////////////////////////////
            ;5517 ;
U 11D6, 0000,003D,5D80,0800,0084,70B2  ;5518 =0 ERROR2[.7]   ; Memory did NOT log MULTIPLE
            ;5519       ; ...TRANSMITTER error
            ;5520 ;
            ;5521 ;////////////////////////////////////////////////////////////////////////////////
            ;5522 ;
            ;5523 ; ERROR LOGOUT:
            ;5524 ;
            ;5525 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5526 ;  NOT USED
            ;5527 ;  CPU ID
            ;5528 ;  EXPECTED STATUS OF ERROR BITS FORM CNFG REG A
            ;5529 ;  RECEIVED STATUS OF ERROR BITS FROM CNFG REG A
            ;5530 ;  RECEIVED CONTENTS OF THE REGISTER
            ;5531 ;  ID COUNTER
            ;5532 ;
            ;5533 ;
            ;5534 ;////////////////////////////////////////////////////////////////////////////////
            ;5535 ;
            ;5536 TEST.8.NXTID2:
U 11D7, 0810,0038,0180,0940,0000,13DB  ;5537  D_RC[8]    ; Get ID Counter
            ;5538  ALU_D[XOR]K[.1F],  ; Is the ID equal to x1F?
U 13DB, 0019,0020,8D80,0800,0010,13DC  ;5539   CLK.UBCC  ; ...
```

```
ZZ-ESKAR-V22  TEST 18E CPU AND MEMORY MULT XMIT FLT                                    SEQ 0353
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23                  Page  16

U 13DC. 0000,013C,0180,0800,0000,11D8  ;5540  Z?     ; ...
        ;5541 =0 RC[8]_D+K[.1],    ; ID Counter not x20 yet.
U 11D8. 0019,0014,0580,09C0,0000,11CC  ;5542   J/TEST.8.LOOP2  ; Repeat the loop
U 11D9. 0000,003E,0180,0800,0000,0001  ;5543  RETURN1     ; Finished the test
        ;5544 TEST.8.EXIT:
U 13DD. 0000,003C,0180,0800,0000,11DA  ;5545  NOP
        ;5546
```

```
            ;5547   .PAGE "TEST 18F RESERVED"
            ;5548 =0
U 11DA, 0000,003D,0180,0800,0000,1063   ;5549 T18F: NEWTST
U 11DB, 0000,003C,0180,0800,0000,11DC   ;5550   NOP
            ;5551
```

```
            ;5552  .PAGE "TEST 190 RESERVED"
            ;5553 =0
U 11DC. 0000,003D,0180,0800,0000,1063  ;5554 T190: NEWTST
U 11DD. 0000,003C,0180,0800,0000,13DE  ;5555  NOP
            ;5556
```

```
            ;5557  .PAGE "DUMMY NEWTST"
U 13DE. 0000,003D,0180,0800,0000,1063  ;5558 DUMMY: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0357
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23     Page  17
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1926: 1906 1915= 1916= 2741= 2742= 2743= 2746=
U 1008   2372= 2374= 2376= 2381= 2361= 2362= 2363= 2364=
U 1010   2395= 2397= 2398= 2402= 2418= 2419= 2420= 2428=
U 1018   2898= 2899= 2900= 2901= 2902= 2903= 2904= 2905=
U 1020   2430= 2432= 2434= 2435= 2478= 2479= 2480= 2482=
U 1028   2526= 2527= 2528= 2530= 3448= 3450= 3454= 3455=
U 1030   3502= 3503= 3504= 1907 3572= 3573= 3578= 1908
U 1038   3660= 3662= 3667= 1909 3765= 3767= 3772= 1911
U 1040   3936= 3938= 3943= 1912 4042= 4044= 4049= 1919
U 1048   4204= 4206= 4211= 1920 1917= 1918= 2025: 1921
U 1050   4306= 4308= 4313= 1922 4453= 4454= 4459= 1923
U 1058   1969= 1971= 2073: 1924 1982= 1983= 2109: 1925
U 1060   4484= 4485= 4490= 1958 4511= 4512= 4517= 1959
U 1068   4569= 4570= 4575= 1960 4595= 4596= 4601= 1961
U 1070   4652= 4653= 4658= 4678= 4725= 4726= 4731= 1962
U 1078   4823= 4824= 4829= 1963 4956= 4957= 4962= 1964
U 1080   4986= 4987= 4992= 1965 5015= 5016= 5021= 1966
U 1088   5045= 5046= 5051= 1967 5102= 5103= 5108= 5130=
U 1090   5133= 5134= 5139= 5157= 5240= 5241= 5242= 1968
U 1098   5247= 5248= 5250= 5251= 5252= 5253= 5254= 1972
U 10A0   5258= 5260= 5261= 5263= 5264= 5265= 5266= 1973
U 10A8   5270= 5276= 5290= 5291= 5292= 5293= 5294= 1974
U 10B0   2043= 2044= 2091= 2092= 2224= 2225= 2350= 2351=
U 10B8   2353= 2358= 2558= 2563= 2568= 2573= 2578= 2579=
U 10C0   2623= 2624= 2701= 2702= 2780= 2781= 2811= 2812=
U 10C8   2877= 2878= 2886= 2887= 2888= 2890= 2893= 2894=
U 10D0   2935= 2936= 2937= 2938= 2945= 2946= 2952= 2954=
U 10D8   2957= 2958= 2977= 2978= 2980= 2981= 3086= 3088=
U 10E0   3091= 3092= 3149= 3151= 3181= 3182= 3185= 3186=
U 10E8   3197= 3198= 3201= 3202= 3215= 3216= 3217= 3219=
U 10F0   3220= 3221= 3222= 3223= 3224= 3225= 3226= 3227=
U 10F8   3228= 3229= 3231= 3232= 3445= 3446= 3510= 3511=
U 1100   1892: 1893: 1894: 1895: 1896: 1897: 1898: 1899:
U 1108   1900: 1975 3512= 3513= 1901: 1902: 1903: 1904:
U 1110   3514= 3515= 3521= 3531= 3537= 3547= 3555= 3568=
U 1118   3606= 3617= 3636= 3637= 3638= 3639= 3640= 3641=
U 1120   3656= 3657= 3693= 3698= 3714= 3719= 3744= 3758=
U 1128   3801= 3815= 3816= 3817= 3825= 3836= 3842= 3854=
U 1130   3861= 3873= 3883= 3910= 3911= 3912= 3913= 3914=
U 1138   3915= 3916= 3932= 3933= 3969= 3974= 3990= 3995=
U 1140   4020= 4035= 4078= 4092= 4093= 4094= 4102= 4114=
U 1148   4120= 4132= 4139= 4151= 4161= 4179= 4180= 4181=
U 1150   4182= 4183= 4184= 4185= 4200= 4201= 4237= 4242=
U 1158   4258= 4263= 4285= 4299= 4342= 4356= 4357= 4358=
U 1160   4366= 4378= 4384= 4396= 4403= 4415= 4425= 4439=
U 1168   4440= 4441= 4442= 4443= 4444= 4445= 4447= 4448=
U 1170   4451= 4452= 4481= 4482= 4546= 4559= 4560= 4561=
U 1178   4567= 4568= 4632= 4645= 4646= 4647= 4685= 4698=
U 1180   4699= 4701= 4709= 4710= 4711= 4712= 4713= 4714=
U 1188   4717= 4718= 4719= 4720= 4722= 4723= 4756= 4761=
U 1190   4776= 4781= 4804= 4818= 4861= 4874= 4875= 4876=
U 1198   4885= 4896= 4902= 4913= 4920= 4945= 4946= 4947=
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 0358
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23           Page  17

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4948= 4949= 4950= 4951= 4952= 4953= 5013= 5014=
U 11A8  5084= 5085= 5086= 5087= 5088= 5089= 5094= 5095=
U 11B0  5096= 5097= 5100= 5101= 5238= 5239= 5309= 5310=
U 11B8  5311= 5312= 5335= 5336= 5339= 5340= 5341= 5342=
U 11C0  5355= 5356= 5367= 5369= 5378= 5396= 5408= 5427=
U 11C8  5432= 5433= 5442= 5443= 5447= 5448= 5449= 5450=
U 11D0  5462= 5463= 5475= 5477= 5489= 5507= 5518= 5537=
U 11D8  5542= 5543= 5549= 5550= 5554= 5555= 1976 1977
U 11E0  1978 1979 1980 1981 2000 2023 2024 2045
U 11E8  2046 2071 2072 2093 2094 2157 2158 2159
U 11F0  2177 2179 2205 2206 2207 2208 2209 2210
U 11F8  2211 2212 2213 2214 2215 2216 2217 2218
U 1200  2219 2220 2222 2223 2359 2360 2385 2405
U 1208  2406 2407 2414 2553 2601 2625 2626 2627
U 1210  2628 2646 2647 2648 2649 2667 2668 2669
U 1218  2670 2696 2730 2732 2733 2734 2736 2737
U 1220  2738 2739 2740 2747 2748 2749 2750 2773
U 1228  2774 2775 2776 2778 2779 2806 2807 2809
U 1230  2810 2838 2839 2840 2879 2880 2881 2882
U 1238  2891 2892 2896 2897 2910 2911 2913 2914
U 1240  2915 2917 2922 2923 2924 2926 2931 2933
U 1248  2934 2942 2943 2947 2948 2949 2950 2951
U 1250  2959 2960 2961 2963 2964 2965 2966 2967
U 1258  2968 2973 2975 2976 2979 3004 3005 3006
U 1260  3007 3035 3036 3038 3039 3040 3080 3081
U 1268  3082 3083 3084 3089 3090 3094 3095 3097
U 1270  3142 3143 3144 3146 3148 3177 3179 3180
U 1278  3184 3188 3189 3190 3191 3193 3194 3195
U 1280  3196 3199 3200 3214 3233 3234 3245 3246
U 1288  3247 3257 3258 3259 3260 3270 3271 3272
U 1290  3273 3283 3284 3285 3286 3295 3296 3297
U 1298  3298 3309 3310 3311 3312 3313 3456 3457
U 12A0  3458 3459 3460 3461 3462 3463 3464 3465
U 12A8  3466 3467 2135: 3468 3469 3470 3471 3472
U 12B0  3473 3474 3475 3476 3477 3478 3479 3480
U 12B8  3481 3482 3483 3484 3485 3486 3487 3488
U 12C0  3489 3490 3491 3492 3493 3494 3495 3496
U 12C8  3497 3498 3516 3517 3532 3549 3550 3570
U 12D0  3571 3598 3599 3600 3601 3642 3643 3644
U 12D8  3645 3646 3647 3648 3649 3650 3651 3652
U 12E0  3653 3654 3658 3659 3689 3690 3691 3692
U 12E8  3712 3713 3734 3735 3736 3737 3738 3739
U 12F0  3759 3760 3761 3762 3763 3764 3794 3795
U 12F8  3796 3818 3819 3820 3837 3855 3856 3874
U 1300  3875 3876 3877 3878 3917 3918 3919 3920
U 1308  3921 3922 3923 3924 3925 3926 3927 3928
U 1310  3929 3930 3934 3935 3965 3966 3967 3968
U 1318  3988 3989 4010 4011 4012 4013 4014 4015
U 1320  4036 4037 4038 4039 4040 4041 4071 4072
U 1328  4073 4095 4096 4097 4115 4133 4134 4152
U 1330  4153 4154 4155 4156 4186 4187 4188 4189
U 1338  4190 4191 4192 4193 4194 4195 4196 4197
U 1340  4198 4202 4203 4233 4234 4235 4236 4256
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4257 4277 4278 4279 4280 4300 4301 4302
U 1350  4303 4304 4305 4335 4336 4337 4359 4360
U 1358  4361 4379 4397 4398 4416 4417 4418 4419
U 1360  4420 4446 4449 4483 4539 4540 454. 4562
U 1368  4563 4564 4565 4623 4625 4626 4627 4651
U 1370  4679 4680 4715 4724 4753 4754 4755 4774
U 1378  4775 4793 4794 4795 4797 4798 4819 4820
U 1380  4821 4822 4851 4852 4853 4854 4855 4856
U 1388  4877 4878 4879 4880 4897 4914 4915 4954
U 1390  5090 5091 5092 5093 5098 5159 5161 5269
U 1398  5313 5314 5315 5317 5318 5319 5320 5321
U 13A0  5322 5323 5324 5325 5343 5344 5346 5347
U 13A8  5349 5350 5351 5352 5354 5357 5358 5359
U 13B0  5360 5361 5362 5363 5365 5366 5372 5373
U 13B8  5399 5400 5402 5403 5429 5430 5444 5451
U 13C0  5452 5453 5455 5456 5458 5459 5460 5461
U 13C8  5464 5465 5466 5467 5468 5469 5470 5471
U 13D0  5473 5474 5479 5480 5481 5483 5484 5509
U 13D8  5511 5513 5514 5539 5540 5545 5558
U 13E0 - 13EF Unused
U 13F0  2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
U 13F8  2127: 2128: 2129: 2130: 2131: 2132: 2133: 2134:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0360
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23     Page  17

            Words not
            in bounds
 ESK00DEF           0
 ESK00MAC           0
 ESKARMAC           0
  ESKAR40        1007

Used                1007
Remaining

Total microwords used in memory U: 1007
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                    SEQ 0361
ESKAR40.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:23  Page  18


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR40.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR41.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 0362

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1910 Micro Monitor Interface Routines
; 1911   Newtest Routine
; 1968   Enable Cache
; 1990   Diseble Cache Routine
; 2012   Set Trap Mask
; 2041   RESET SUBTEST NUMBER
; 2064   ERLOOP ROUTINE
; 2076   ERROR1 ROUTINE
; 2089   ERROR1 WITH PARAMETER
; 2111   ERROR 2 ROUTINE
; 2124   ERROR 2 WITH PARAMETER
; 2145   CONSOLE CALL ROUTINE
; 2159   WCS SCRATCH PAD LOCATIONS
; 2190   SUBTEST ROUTINE
; 2207   SET ARGUMENT COUNT ROUTINE
; 2224   CLEAR SBI.ERR REGISTER ROUTINE
; 2239   Initialize the SBI
; 2267   SBI UNJAM SUBROUTINE
; 2296   CPUCA ROUTINE
; 2330   SILO UNLOCK AND CLEAR
; 2350   LOAD RC'S WITH SILO DATA
; 2385   GET EXPECTED SILO DATA TO D REG
; 2407   SET SIL COMP FOR UNCOND LOCK
; 2428   GENERATE CONSTANTS SUBROUTINE
; 2473   GET SIL COMP REG TO D FOR LOCK ON ID CYCLE
; 2494   GET SIL COMP REG DATA IN D FOR ID COMPARE LOCK
; 2512   SET MAINT ID AND TURN OFF CACHE
; 2536   SET SIL COMP REG TO LOCK ON ID AND TAG COMPARE
; 2560   SET SIL COMP REG TO LOCK ON ID, TAG AND FUNC
; 2584   FIND MS780-E MEMORY
; 2726   Generate IO Address
; 2752   Set Starting Address
; 2786   Select Controller
; 2824   SELECT LOWER CONTROLLER
; 2873   SELECT UPPER CONTROLLER
; 2921   ENABLE NEXT MS780-E
; 2973   LOAD TEMPORARY REGISTERS
; 3015   CONFIGURE MS780-E/H
; 3088 TEST 191 DATA INTEGRITY OF ID REG 1C
; 3222 TEST 192 COMPARATOR REGISTER COUNT FIELD TEST
; 3319 TEST 193 SILO LOCK ON ID COMPARE TEST
; 3526 TEST 194 SILO LOCK ON ID & TAG COMPARE TEST
; 3806 TEST 195 SILO LOCK ON ID,TAG & FUNCTION COMPARE TEST
; 4069 TEST 196 RESERVED
; 4074 TEST 197 RESERVED
; 4079 TEST 198 RESERVED
; 4084 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR41
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7001  ;1862 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7001  ;1863 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7001  ;1864 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7001  ;1865 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7001  ;1866 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7001  ;1867 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7001  ;1868 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7001  ;1869 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7001  ;1870 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D. 0000.003C.8980.0800.0084.7001  ;1871 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7001  ;1872 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1017  ;1874 TRPH0: Q_ID[USTACK]
U 1017. 0C00.003C.01E0.0800.0000.101B  ;1875  Q_D,D_Q
U 101B. 0000.003C.8180.3C00.0000.101F  ;1876  ID[USTACK]_D
U 101F. 0C00.003C.01E0.0800.0000.104F  ;1877  Q_D,D_Q
U 104F. 0000.003C.01C0.0A78.0000.105B  ;1878  Q_R[0F]
U 105B. 0001.2024.0180.0800.0010.105F  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 105F. 0000.013C.0180.0800.0000.1002  ;1880  Z?    ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1024   ;1897 TRPH1: RC[0D]_SC
U 1024. 0000.003D.D980.0800.0084.71A4   ;1898 =0 SETRCF[.9]
U 1025. 0000.003C.49F0.2C00.0000.1109   ;1899  Q_ID[TBER0]
U 1109. 0001.203C.4DF0.2DB0.0000.1154   ;1900  RC[6]_Q,Q_ID[TBER1]
U 1154. 0001.203C.65F0.2DB8.0000.1177   ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 1177. 0001.203C.6DF0.2DD8.0000.1178   ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1178. 0001.203C.75F0.2DE0.0000.1179   ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1179. 0001.203C.79F0.2DC8.0000.117A   ;1904  RC[9]_Q,Q_ID[PARITY]
U 117A. 0001.203C.69F0.2DC0.0000.117B   ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 117B. 0001.203C.81F0.2DD0.0000.1000   ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.119E   ;1907 1000: RC[0E]_Q,ERROR1
            ;1908
```

```
                ;1909   .PAGE
                ;1910   .TOC " Micro Monitor Interface Routines"
                ;1911   .TOC "   Newtest Routine"
                ;1912 ;**
                ;1913 ; FUNCTIONAL DESCRIPTION:
                ;1914 ;
                ;1915 ; This routine Initializes the following registers:
                ;1916 ;  PSL    to 1F
                ;1917 ;  CES    to 0
                ;1918 ;  ACC.CS    to disable accellerator
                ;1919 ;  SIR    to 0
                ;1920 ;  CLK.CS    to stop interval timer
                ;1921 ;  TBER0    to disable the TB
                ;1922 ;  SBI.MAINT to     0
                ;1923 ;  SBI.ERR    to 0
                ;1924 ;  FAULT    to 0
                ;1925 ;  COMP    to 0
                ;1926 ;  RC[0E]    to  0
                ;1927 ;  R[0F]    to 0
                ;1928 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1929 ; A SCOPE call is then made to the Monitor.
                ;1930 ;
                ;1931 ; CALLING CONSTRAINT:
                ;1932 ;
                ;1933 ; =0
                ;1934 ;
                ;1935 ; SIDE EFFECTS:
                ;1936 ;
                ;1937 ; D Reg is destroyed
                ;1938 ; SC is destroyed
                ;1939 ;--
U 117C. 0000.003C.65F8.0800.0084.717D   ;1940 SCOPE: sc_k[.10],q_0   ; Set IPL to 1F
U 117D. 0818.0038.8D80.0800.0000.117E   ;1941  d_k[.1f]   ; D=1F
U 117E. 0D00.003C.0180.0800.0000.117F   ;1942  d_dal.sc   ; D=001F0000
U 117F. 0000.003C.3D80.3C00.0000.1180   ;1943  id[psl]_d   ; psl = 001F0000
U 1180. 0818.0038.0580.0800.0000.1181   ;1944  d_k[.1]    ; Clear the CES register
U 1181. 0D00.003C.0180.0800.0000.1182   ;1945  d_dal.sc   ; D = 00010000
U 1182. 0F00.003C.3180.3C00.0000.1183   ;1946  id[ces]_d,d_0   ; ces = 00010000
U 1183. 0000.003C.5D80.3C00.0000.1184   ;1947  id[acc.cs]_d   ; acc.cs = 0
U 1184. 0000.003C.3980.3C00.0000.1185   ;1948  id[sir]_d   ; sir = 0
U 1185. 0000.003C.2980.3C00.0000.1186   ;1949  id[clk.cs]_d   ; clk.cs = 0
U 1186. 0000.003C.4980.3C00.0000.1026   ;1950  id[tber0]_d   ; tber0 = 0
U 1026. 0000.003D.0180.0800.0000.11AE   ;1951 =0 call[sbi.unjam]   ; Unjam the SBI
                ;1952   intrpt.strobe,   ; Strobe interrupts
U 1027. 0818.0038.C180.0800.4000.1187   ;1953       d_k[.ffff]   ; Setup to clear SBI error register and
U 1187. 0801.0028.6580.3C00.0000.1188   ;1954  id[sbi.err]_d,d_not.d ; and fault register
U 1188. 0F00.003C.6D80.3C00.0000.1189   ;1955  id[fault]_d,d_0   ; ...
U 1189. 0000.003C.6D80.3C00.0000.118A   ;1956  id[fault]_d   ; fault = 0
U 118A. 0000.003C.7580.3C00.0000.118B   ;1957  id[maint]_d   ; MAINT = 0
U 118B. 0000.003C.6580.3C00.0000.118C   ;1958  id[sbi.err]_d   ; sbi error reg = 0
U 118C. 0000.003C.7180.3C00.0000.118D   ;1959  id[comp]_d   ; comp reg = 0
U 118D. 0000.003C.E580.3C00.0000.118E   ;1960  id[T9]_d   ; Clear code for subroutine START.TEST
U 118E. 0001.003C.0180.09F0.0000.118F   ;1961  rc[0e]_d   ;
U 118F. 0001.003C.0180.0AF8.0000.1190   ;1962  r[0f]_d    ; Clear expected trap mask
U 1190. 0001.003C.0180.09F8.0000.1028   ;1963  rc[0f]_d   ; Clear subtest number and argumnet count
```

```
U 1028, 0000,003D,0180,0800,0000,1195  ;1964 =0 call[disable.cache] ; Disable the cache
U 1029, 0F03,003C,0180,09E8,0000,105E  ;1965  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1966
```

```
            ;1967  .PAGE
            ;1968  .TOC "   Enable Cache"
            ;1969 ;++
            ;1970 ; FUNCTIONAL DESCRIPTION:
            ;1971 ;
            ;1972 ; This routine enables cache group 0 by clearing the force
            ;1973 ; miss bit in the maintenance register.
            ;1974 ;
            ;1975 ; CALLING CONSTRAINT:
            ;1976 ;
            ;1977 ; =0
            ;1978 ;
            ;1979 ; SIDE EFFECTS:
            ;1980 ;
            ;1981 ; D, Q AND SC Registers destroyed
            ;1982 ;--
            ;1983 ENABLE.CACHE:
U 1191, 0000.003C.75F0.2C00.0000.1192  ;1984  q_id[maint]  ; Get current maintenance register
U 1192, 0000.003C.6580.0800.0084.7193  ;1985  sc_k[.10]  ; Setup to clear bit 16
U 1193, 0801.2034.0180.0800.0000.1194  ;1986  d_q.and.mask  ; Clear bit 16
U 1194, 0000.003E.7580.3C00.0000.0001  ;1987  id[maint]_d,return1 ; Rewrite register and return
            ;1988
```

```
            ;1989  .PAGE
            ;1990  .TOC "   Disable Cache Routine"
            ;1991 ;**
            ;1992 ; FUNCTIONAL DESCRIPTION:
            ;1993 ;
            ;1994 ; This routine disables cache hits by setting bits <16:15>
            ;1995 ; in the maintenance register.
            ;1996 ;
            ;1997 ; CALLING SEQUENCE:
            ;1998 ;
            ;1999 ; =0
            ;2000 ;
            ;2001 ; SIDE EFFECTS:
            ;2002 ;
            ;2003 ; D & Q Registers destroyed
            ;2004 ;--
            ;2005 DISABLE.CACHE:
U 1195, 081B.0038.4580.0800.0000.1196  ;2006  d_k[.8000]  ; ... and seed constant
U 1196, 0500.003C.0180.0800.0000.1197  ;2007  d_d.left  ; Generate final constant
U 1197, 0819.0030.4580.0800.0000.1198  ;2008  d_d.or.k[.8000]  ; ... = 00018000
U 1198, 0000.003E.7580.3C00.0000.0001  ;2009  id[maint]_d,return1 ; Disable cache and return
            ;2010
```

```
            ;2011   .PAGE
            ;2012   .TOC "   Set Trap Mask"
            ;2013 ;**
            ;2014 ; FUNCTIONAL DESCRIPTION:
            ;2015 ;
            ;2016 ; This routine is used to set a bit in the Micro Trap Mask
            ;2017 ; R[0F].
            ;2018 ;
            ;2019 ; CALLING CONSTRAINT:
            ;2020 ;
            ;2021 ; =0
            ;2022 ;
            ;2023 ; INPUTS:
            ;2024 ;
            ;2025 ; SC - Contains bit number to set.
            ;2026 ;
            ;2027 ; OUTPUTS:
            ;2028 ;
            ;2029 ; rc[0E] - Contains the current trap mask
            ;2030 ;
            ;2031 ; SIDE EFFECTS:
            ;2032 ;
            ;2033 ; d regsiter is destroyed
            ;2034 ;--
            ;2035 SET_TRAP_MASK:
U 1199, 0800,003C,0180,0A78,0000,119A  ;2036  d_r[0f]
U 119A, 0801,001C,0180,0AF8,0000,119B  ;2037  r[0f]_d.ornot.mask,d_alu ; Set mask
U 119B, 0001,003E,0180,0AF0,0000,0001  ;2038  r[0E]_d,return1    ; Save mask and return
            ;2039
```

```
            ;2040  .PAGE
            ;2041  .TOC "   RESET SUBTEST NUMBER"
            ;2042 ;**
            ;2043 ; FUNCTIONAL DESCRIPTION:
            ;2044 ;
            ;2045 ; Since some  of the tests will have to be restarted when two
            ;2046 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2047 ; the subtest counter to zero ( only for thoes tests that have
            ;2048 ; more than one subtest). This subroutine may also be necessary
            ;2049 ; when a test is being loop on.
            ;2050 ;
            ;2051 ; CALLING CONSTRAINT:
            ;2052 ;
            ;2053 ; =0
            ;2054 ; SIDE EFFECTS:
            ;2055 ;
            ;2056 ; D is destroyed
            ;2057 ;
            ;2058 ;--
            ;2059 RESET.SUBTST:
            ;2060  RC[0F]_0,    ; Reset subtest number
U 119C, 0003,003E,0180,09F8,0000,0001  ;2061   RETURN1    ; ...and return
            ;2062
```

```
            ;2063  .PAGE
            ;2064  .TOC "   ERLOOP ROUTINE"
            ;2065 ;
            ;2066 ;*****************************************************************************
            ;2067 ;**
            ;2068 ;
            ;2069 ;
            ;2070 ;--
            ;2071 ;*****************************************************************************
            ;2072 ;
U 119D. 0818,0038,0980,0800,0000,105E  ;2073 ERLOOP: D_K[.21,J/CALLCON  ; ERRLOOP CALL
            ;2074
```

```
            ;2075  .PAGE
            ;2076  .TOC "   ERROR1 ROUTINE"
            ;2077 ;
            ;2078 ;=============================================================================
            ;2079 ;++
            ;2080 ;
            ;2081 ;
            ;2082 ;--
            ;2083 ;=============================================================================
            ;2084 ;
U 119E. 0810.0038.0180.0978.0000.119F  ;2085 ERROR1: D_RC[0F]
U 119F. 0819.0034.4D80.0800.0000.104E  ;2086  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;2087 104E: D_D.OR.K[.4],J/CALLCON
```

```
            ;2088  .PAGE
            ;2089  .TOC "   ERROR1 WITH PARAMETER"
            ;2090 ;**
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This subroutine takes  as parameter the number of arguments
            ;2094 ; to be printed to the console in the case of an error logout.
            ;2095 ;
            ;2096 ; CALLING CONSTRAINT:
            ;2097 ;
            ;2098 ; =0
            ;2099 ;INPUTS:
            ;2100 ;
            ;2101 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2102 ;--
            ;2103 =0
            ;2104 ERR1.ARG:
U 102A, 0000.003D.0180.0800.0000.11A4  ;2105  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 102B, 0000.003C.0180.0800.0000.119E  ;2106  J/ERROR1  ; Go to ERROR1
            ;2107 ;
            ;2108 ;
            ;2109
```

```
            ;2110  .PAGE
            ;2111  .TOC "    ERROR 2 ROUTINE"
            ;2112 ;
            ;2113 ;*****************************************************************************
            ;2114 ;++
            ;2115 ;
            ;2116 ;
            ;2117 ;--
            ;2118 ;*****************************************************************************
            ;2119 ;
U 11A0, 0810.0038.0180.0978.0000.11A1  ;2120 ERROR2: D_RC[0F]
U 11A1, 0819.0034.4D80.0800.0000.105A  ;2121  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;2122 105A: D_D.OR.K[.6],J/CALLCOM
```

```
            ;2123  .PAGE
            ;2124  .TOC "   ERROR 2 WITH PARAMETER"
            ;2125 ;++
            ;2126 ; FUNCTIONAL DESCRIPTION:
            ;2127 ;
            ;2128 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2129 ; except that in this case after setting the number of arguments
            ;2130 ; too the console, control is transferred to routine ERROR2.
            ;2131 ;
            ;2132 ; INPUTS:
            ;2133 ;
            ;2134 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2135 ;
            ;2136 ;--
            ;2137 =0
            ;2138 ERR2.ARG:
U 102C, 0000,003D,0180,0800,0000,11A4  ;2139  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 102D, 0000,003C,0180,0800,0000,11A0  ;2140  J/ERROR2  ; Go to ERROR2
            ;2141 ;
            ;2142 ;
            ;2143
```

```
          ;2144  .PAGE
          ;2145  .TOC "    CONSOLE CALL ROUTINE"
          ;2146 ;
          ;2147 ;=========================================================================
          ;2148 ;++
          ;2149 ;
          ;2150 ;
          ;2151 ;--
          ;2152 ;=========================================================================
          ;2153 ;
          ;2154 105E:
          ;2155 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2156  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
          ;2157
```

```
            :2158  .PAGE
            :2159  .TOC "    WCS SCRATCH PAD LOCATIONS"
            :2160 ;
            :2161 ;*****************************************************************************
            :2162 ;**
            :2163 ;
            :2164 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            :2165 ;
            :2166 ;
            :2167 ;--
            :2168 ;*****************************************************************************
            :2169 ;
U 13F0, 0000.003C.0180.0800.0000.13F1   ;2170 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2   ;2171 13F1: NOP
U 13F2, 0000.003C.0180.0800.0000.13F3   ;2172 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4   ;2173 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5   ;2174 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6   ;2175 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7   ;2176 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8   ;2177 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9   ;2178 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA   ;2179 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB   ;2180 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC   ;2181 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD   ;2182 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE   ;2183 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF   ;2184 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155   ;2185 13FF: NOP
U 1155, 0000.003C.0180.0800.0000.12AA   ;2186 1155: NOP
U 12AA, 0000.003C.0180.0800.0000.11A2   ;2187 12AA: NOP
            ;2188
```

```
            ;2189  .PAGE
            ;2190  .TOC "   SUBTEST ROUTINE"
            ;2191 ;
            ;2192 ;*****************************************************************************
            ;2193 ;**
            ;2194 ;
            ;2195 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2196 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2197 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2198 ; TYPING IT.
            ;2199 ;
            ;2200 ;--
            ;2201 ;*****************************************************************************
            ;2202 ;
U 11A2, 0810,0038,4D80,0978,0000,11A3  ;2203 SUBTST: D_RC[0F],KMX/.FF00
U 11A3, 0019,4016,4D80,09F8,0000,0001  ;2204  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2205
```

```
            ;2206  .PAGE
            ;2207  .TOC "   SET ARGUMENT COUNT ROUTINE"
            ;2208 ;
            ;2209 ;*****************************************************************************
            ;2210 ;**
            ;2211 ;
            ;2212 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2213 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2214 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2215 ;
            ;2216 ;--
            ;2217 ;*****************************************************************************
            ;2218 ;
U 11A4. 0010,0038,01C0,0978,0000,11A5  ;2219 SETRCF: Q_RC[0F]
U 11A5. 0019,2034,4DC0,0800,0000,11A6  ;2220  Q_Q.AND.K[.FF00]
U 11A6. 0019,2032,1D80,09F8,0000,0001  ;2221  RC[0F]_Q.OR.SC,RETURN1
            ;2222
```

```
            ;2223  .PAGE
            ;2224  .TOC "    CLEAR SBI.ERR REGISTER ROUTINE"
            ;2225 ;
            ;2226 ;******************************************************************************
            ;2227 ;++
            ;2228 ;
            ;2229 ; THIS ROUTINE CLEARS THE SBI.ERR REGISTER
            ;2230 ;
            ;2231 ;--
            ;2232 ;******************************************************************************
            ;2233 ;
U 11A7, 081B.0000.0580.0800.0000.11A8  ;2234 CLCPTO: D_0-K[.1]
U 11A8, 0801.0028.6580.3C00.0000.11A9  ;2235  ID[SBI.ERR]_D,D_NOT.D
U 11A9, 0000.003E.6580.3C00.0000.0001  ;2236  ID[SBI.ERR]_D,RETURN1
            ;2237
```

```
            ;2238  .PAGE
            ;2239  .TOC "   Initialize the SBI"
            ;2240 ;++
            ;2241 ; FUNCTIONAL DESCRIPTION:
            ;2242 ;
            ;2243 ; This routine performs the following functions:
            ;2244 ;
            ;2245 ;  Executes an SBI Unjam
            ;2246 ;  Clears the SBI Fault Latch
            ;2247 ;  Clears the SBI Error Register
            ;2248 ;
            ;2249 ; CALLING CONSTRAINT:
            ;2250 ;
            ;2251 ; =0
            ;2252 ;
            ;2253 ; SIDE EFFECTS:
            ;2254 ;
            ;2255 ; D & Q Register destroyed
            ;2256 ;--
            ;2257 =0
            ;2258 SBI.INIT:
U 102E. 0000.003D.0180.0800.0000.11AE  ;2259  call[sbi.unjam]  ; Unjam the SBI
U 102F. 0818.0038.C180.0800.0000.11AA  ;2260  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11AA. 0801.0028.6580.3C00.0000.11AB  ;2261  id[sbi.err]_d,d_not.d ; Clear all write/one/to/clear bits
U 11AB. 0F00.003C.6D80.3C00.0000.11AC  ;2262  id[fault]_d,d_0
U 11AC. 0000.003C.6D80.3C00.0000.11AD  ;2263  id[fault]_d
U 11AD. 0000.003E.6580.3C00.0000.0001  ;2264  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2265
```

```
            ;2266  .PAGE
            ;2267  .TOC "    SBI UNJAM SUBROUTINE"
            ;2268 ;
            ;2269 ;*****************************************************************************
            ;2270 ;**
            ;2271 ;
            ;2272 ;     THIS ROUTINE PERFORMS AN UNJAM SEQUENCE ON THE SBI.
            ;2273 ;
            ;2274 ;--
            ;2275 ;*****************************************************************************
            ;2276 ;
            ;2277 SBI.UNJAM:
            ;2278 HLD.UNJ:
U 11AE. 0818.0038.1180.8000.0000.11AF  ;2279  MCT/SBI.HOLD,D_K[.4] ; ASSERT HOLD FOR 16 CYCLES
U 11AF. 0001.003C.0180.8000.0010.11B0  ;2280 HLD0: MCT/SBI.HOLD,ALU_D,CLK.UBCC
U 11B0. 0819.0100.0580.8000.0000.1030  ;2281  MCT/SBI.HOLD,Z?,D_D-K[.1]
U 1030. 0000.003C.0180.8000.0000.11AF  ;2282 =0 MCT/SBI.HOLD,J/HLD0
U 1031. 0000.003C.0180.8000.0000.11B1  ;2283  MCT/SBI.HOLD
U 11B1. 0818.0038.1180.8800.0000.11B2  ;2284 UNJAM1: MCT/SBI.HOLD+UNJAM,D_K[.4] ; ASSERT HOLD+UNJAM FOR 16 CYCLES
U 11B2. 0001.003C.0180.8800.0010.11B3  ;2285 HLD1: MCT/SBI.HOLD+UNJAM,ALU_D,CLK.UBCC
U 11B3. 0819.0100.0580.8800.0000.1032  ;2286  MCT/SBI.HOLD+UNJAM,Z?,D_D-K[.1]
U 1032. 0000.003C.0180.8800.0000.11B2  ;2287 =0 MCT/SBI.HOLD+UNJAM,J/HLD1
U 1033. 0000.003C.0180.8800.0000.11B4  ;2288  MCT/SBI.HOLD+UNJAM
U 11B4. 0818.0038.1180.8000.0000.11B5  ;2289 UNJAM2: MCT/SBI.HOLD,D_K[.4] ; ASSERT HOLD FOR 16 CYCLES
U 11B5. 0001.003C.0180.8000.0010.11B6  ;2290 HLD2: MCT/SBI.HOLD,ALU_D,CLK.UBCC
U 11B6. 0819.0100.0580.8000.0000.1034  ;2291  MCT/SBI.HOLD,Z?,D_D-K[.1]
U 1034. 0000.003C.0180.8000.0000.11B5  ;2292 =0 MCT/SBI.HOLD,J/HLD2
U 1035. 0000.003E.0180.8000.4000.0001  ;2293  IEK/ISTR,MCT/SBI.HOLD,RETURN1
            ;2294
```

```
            ;2295  .PAGE
            ;2296  .TOC "   CPUCA ROUTINE"
            ;2297 ;
            ;2298 ;*******************************************************************************
            ;2299 ;**
            ;2300 ;
            ;2301 ;       THIS ROUTINE PLACES THE EXPECTED DATA PATTERN FOR
            ;2302 ; CPU COMMAND ADDRESS AS IT APPEARS IN THE SILO.
            ;2303 ; IT GENERATES THE ID AND TAG FIELDS ONLY, I.E. IT IS
            ;2304 ; UP TO THE CALLING ROUTINE TO "OR IN" THE FUNCTION CODE.
            ;2305 ;
            ;2306 ; D IS RETURNED WITH THE VALUE 00XX,XXX0,1100,XXXX,XXXX,XXXX,XXXX
            ;2307 ;  X'S ARE DEPENDENT ON CPU'S ID AND TR LEVEL
            ;2308 ; SC , Q,RC[0E]-RC[0] GET CLOBBERED
            ;2309 ;
            ;2310 ;--
            ;2311 ;*******************************************************************************
            ;2312 ;
            ;2313 =0
U 1036. 0000.003C.8980.0800.0084.7037   ;2314 CPUCA: SC_K[.D]
U 1037. 0003.001C.0180.0AF8.0000.1038   ;2315  R[0F]_NOT.MASK
U 1038. 0000.003D.0180.0800.0000.11BA   ;2316 =0 CALL,J/SILCLR
U 1039. 0818.0038.1980.0800.0000.103A   ;2317  D_K[ZERO]    ; Get TR level
U 103A. 0000.003D.0180.0800.0000.122C   ;2318 =0 CALL[GENERATE.ID]  ; Generate IO Address based
            ;2319       ; ...on TR level
U 103B. 0001.003C.7180.3C00.0200.103C   ;2320  ID[COMP]_D.VA_D
U 103C. 0000.003D.0180.C800.0000.1042   ;2321 =0 CALL,J/SLOLCK,MCT/READ.P
U 103D. 0000.003C.0180.0800.0000.103E   ;2322  NOP
U 103E. 0000.003D.0180.0800.0000.11A7   ;2323 =0 CALL,J/CLCPTO
U 103F. 0810.0038.0180.0958.0000.11B7   ;2324  D_RC[0B]
U 11B7. 0000.003C.2180.0800.0084.71B8   ;2325  SC_K[.14]
U 11B8. 0000.003C.0980.0800.0084.B1B9   ;2326  SC_SC-K[.2]
U 11B9. 0801.0036.0180.0800.0000.0001   ;2327  ALU_D.AND.MASK,D_ALU,RETURN1
            ;2328
```

```
            ;2329  .PAGE
            ;2330  .TOC "    SILO UNLOCK AND CLEAR"
            ;2331 ;
            ;2332 ;*****************************************************************************
            ;2333 ;**
            ;2334 ;
            ;2335 ; THIS ROUTINE UNLOCKS THE SILO,AND WAITS FOR 16 CYCLES
            ;2336 ; TO CLEAR IT OUT.
            ;2337 ;
            ;2338 ;--
            ;2339 ;*****************************************************************************
            ;2340 ;
U 11BA. 0F00.003C,0180.0800,0000.11BB  ;2341 SILCLR: D_0
U 11BB. 0000.003C,7180.3C00,0000.11BC  ;2342  ID[COMP]_D
U 11BC. 081B.0038,1180.0800,0000.11BD  ;2343  D_K[.4]
U 11BD. 0001.003C,0180.0800,0010.11BE  ;2344 CLR0: ALU_D,CLK.UBCC
U 11BE. 0000.013C,0180.0800,0000.1040  ;2345  Z?
U 1040. 0819.0000,0580.0800,0000.11BD  ;2346 =0 J/CLR0,D_D-K[.1]
U 1041. 0000.003E,0180.0800,0000.0001  ;2347  RETURN1
            ;2348
```

```
            ;2349  .PAGE
            ;2350  .TOC "   LOAD RC'S WITH SILO DATA"
            ;2351 ;
            ;2352 ;=============================================================================
            ;2353 ;**
            ;2354 ;
            ;2355 ;      THIS ROUTINE PLACES THE LAST 15 LOCATIONS OF THE SILO
            ;2356 ; IN RC[0E] THRU RC[0].
            ;2357 ;
            ;2358 ;  THE STATE IN WHICH THE CALL IS DONE, AND
            ;2359 ;  THE PRECEDING STATE MUST LOOK EXACTLY AS FOLLOWS:
            ;2360 ;
            ;2361 ;   ID[COMP]_D,...
            ;2362 ;   CALL,J/SLOLCK,MCT/XXXXX
            ;2363 ;
            ;2364 ;   WHERE XXXXX IS ANY REFERENCE.
            ;2365 ;
            ;2366 ;--
            ;2367 ;=============================================================================
            ;2368 ;
            ;2369 =0
U 1042, 0818,0038,1180,0800,0000,1043  ;2370 SLOLCK: D_K[.4]    ; PROCEED TO WAIT A TOTAL OF 16 CYCLES
U 1043, 0001,003C,0180,0800,0010,11BF  ;2371 L1: ALU_D,CLK,UBCC  ; ...
U 11BF, 0819,0100,0580,0800,0000,1044  ;2372  Z?,D_D-K[.1]
U 1044, 0000,003C,0180,0800,0000,1043  ;2373 =0 J/L1    ; ...
U 1045, 0000,003C,6180,0800,0084,71C0  ;2374  SC_K[.F]   ; START FOR PLACING SILO IN RC REGISTERS
U 11C0, 0000,003C,0580,0800,0084,B1C1  ;2375 L2: SC_SC-K[.1]   ; STARTING WITH RC[0E]
U 11C1, 0000,003C,61F0,2C00,0000,11C2  ;2376  Q_ID[SILO]
U 11C2, 0001,203C,0180,0838,0000,11C3  ;2377  RC(SC)_ALU,ALU_Q
U 11C3, 0018,0038,1D80,0800,0010,11C4  ;2378  ALU_SC,CLK,UBCC
U 11C4, 0000,013C,0180,0800,0000,1046  ;2379  Z?
U 1046, 0000,003C,0180,0800,0000,11C0  ;2380 =0 J/L2
U 1047, 0F00,003C,0180,0800,0000,11C5  ;2381  D_0
U 11C5, 0000,003E,7180,3C00,0000,0001  ;2382  ID[COMP]_D,RETURN1 ; UNLOCK SILO
            ;2383
```

```
            ;2384  .PAGE
            ;2385  .TOC "   GET EXPECTED SILO DATA TO D REG"
            ;2386 ;
            ;2387 ;*******************************************************************************
            ;2388 ;**
            ;2389 ;
            ;2390 ; THIS ROUTINE PUTS THE EXPECTED DATA FOR THE SILO COMPARATOR
            ;2391 ; REGISTER (AFTER LOCK BIT <31> ASSERTS) INTO THE D REGISTER.
            ;2392 ;
            ;2393 ;--
            ;2394 ;*******************************************************************************
            ;2395 ;
U 11C6. 0818.0038.6180.0800.0000.11C7  ;2396 LKCYC:  D_K[.F]
U 11C7. 0000.003C.65F8.0800.0084.71C8  ;2397  SC_K[.10],Q_0
U 11C8. 0D00.003C.0180.0800.0000.11C9  ;2398  D_BAL.SC    ;SET COUNT FIELD = F
U 11C9. 0000.003C.5180.0800.0084.71CA  ;2399  SC_K[.1E]
U 11CA. 0000.003C.0580.0800.0084.B1CB  ;2400  SC_SC-1
U 11CB. 0801.001C.0180.0800.0000.11CC  ;2401  D_D.ORNOT.MASK    ;SET BIT <29>,UNCOND LOCK
U 11CC. 0000.003C.8D80.0800.0084.71CD  ;2402  SC_K[.1F]
U 11CD. 0801.001C.0180.0800.0000.11CE  ;2403  D_D.ORNOT.MASK  ;SET BIT <31>,SILO LOCK BIT
U 11CE. 0000.003E.0180.0800.0000.0001  ;2404  RETURN1
            ;2405
```

```
            ;2406  .PAGE
            ;2407  .TOC "   SET SIL COMP FOR UNCOND LOCK"
            ;2408 ;
            ;2409 ;*****************************************************************************
            ;2410 ;**
            ;2411 ;
            ;2412 ; THIS ROUTINE SETS UP THE SILO COMPARATOR REGISTER
            ;2413 ; DATA FOR AN UNCONDITIONAL LOCK. THE COUNT FIELD TO
            ;2414 ; BE USED MUST BE IN R3 <3:0>. THE D REGISTER
            ;2415 ; IS RETURNED WITH THE DATA.
            ;2416 ;--
            ;2417 ;*****************************************************************************
            ;2418 ;
U 11CF. 0800.003C.0180.0A18.0000.11D0  ;2419 COMDAT:D_R[03]
U 11D0. 0000.003C.65F8.0800.0084.71D1  ;2420  SC_K[.10],Q_0
U 11D1. 0D00.003C.0180.0800.0000.11D2  ;2421  D_DAL.SC    ;PUT COUNT FIELD IN <19:16>
U 11D2. 0018.0038.51C0.0800.0000.11D3  ;2422  Q_K[.1E]
U 11D3. 0019.2000.0580.0800.0082.11D4  ;2423  SC_Q-K[.1]
U 11D4. 0801.001C.0180.0800.0000.11D5  ;2424  D_D.ORNOT.MASK    ;SET <29>, UNCOND LOCK
U 11D5. 0000.003E.0180.0800.0000.0001  ;2425  RETURN1
            ;2426
```

```
            :2427  .PAGE
            :2428  .TOC "    GENERATE CONSTANTS SUBROUTINE"
            :2429 ;
            :2430 ;*******************************************************************************
            :2431 ;**
            :2432 ;
            :2433 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            :2434 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            :2435 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            :2436 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            :2437 ;
            :2438 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG
            :2439 ; THEN THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            :2440 ;
            :2441 ;   =0  D_0,CALL,J/DC5
            :2442 ;     NOP
            :2443 ;   =0  CALL,J/DC5
            :2444 ;
            :2445 ;--
            :2446 ;*******************************************************************************
            :2447 ;
U 11D6.  0018.0038.1980.0800.0000.11E6   :2448 DC0: ALU_0(K),J/S00
U 11D7.  0018.0038.1980.0800.0000.11E7   :2449 DC1: ALU_0(K),J/S01
U 11D8.  0018.0038.1980.0800.0000.11E8   :2450 DC2: ALU_0(K),J/S10
U 11D9.  0018.0038.1980.0800.0000.11E9   :2451 DC3: ALU_0(K),J/S11
U 11DA.  0018.0038.0980.0800.0000.11E6   :2452 DC4: ALU_K[.2],J/S00
U 11DB.  0018.0038.0980.0800.0000.11E7   :2453 DC5: ALU_K[.2],J/S01
U 11DC.  0018.0038.0980.0800.0000.11E8   :2454 DC6: ALU_K[.2],J/S10
U 11DD.  0018.0038.0980.0800.0000.11E9   :2455 DC7: ALU_K[.2],J/S11
U 11DE.  0018.0038.0580.0800.0000.11E6   :2456 DC8: ALU_K[.1],J/S00
U 11DF.  0018.0038.0580.0800.0000.11E7   :2457 DC9: ALU_K[.1],J/S01
U 11E0.  0018.0038.0580.0800.0000.11E8   :2458 DCA: ALU_K[.1],J/S10
U 11E1.  0018.0038.0580.0800.0000.11E9   :2459 DCB: ALU_K[.1],J/S11
U 11E2.  0018.0038.C180.0800.0000.11E6   :2460 DCC: ALU_K[.FFFF],J/S00
U 11E3.  0018.0038.C180.0800.0000.11E7   :2461 DCD: ALU_K[.FFFF],J/S01
U 11E4.  0018.0038.C180.0800.0000.11E8   :2462 DCE: ALU_K[.FFFF],J/S10
U 11E5.  0018.0038.C180.0800.0000.11E9   :2463 DCF: ALU_K[.FFFF],J/S11
            :2464
U 11E6.  0118.0038.1B80.0800.0000.11EA   :2465 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 11E7.  0118.0038.0B80.0800.0000.11EA   :2466 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 11E8.  0118.0038.0780.0800.0000.11EA   :2467 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 11E9.  0118.0038.C380.0800.0000.11EA   :2468 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
            :2469
U 11EA.  0100.003E.0380.0800.0000.0001   :2470 SX: D_D.LEFT2,SI/7,RETURN1   ;EXIT
            :2471
```

```
            ;2472  .PAGE
            ;2473  .TOC "   GET SIL COMP REG TO D FOR LOCK ON ID CYCLE"
            ;2474 ;
            ;2475 ;*****************************************************************************
            ;2476 ;**
            ;2477 ;
            ;2478 ; THIS ROUTINE PUTS THE EXPECTED SILO COMPARATOR REGISTER
            ;2479 ; DATA FOR A CYCLE THAT LOCKED ON ID ONLY INTO
            ;2480 ; THE D REGISTER.
            ;2481 ;
            ;2482 ;--
            ;2483 ;*****************************************************************************
            ;2484 ;
U 11EB. 0818.0038.4180.0800.0000.1048  ;2485 EXCOMP: D_K[.80]
U 1048. 0000.003D.0180.0800.0000.11E5  ;2486 =0 CALL.J/DCF
U 1049. 0000.003C.65F8.0800.0084.71EC  ;2487  SC_K[.10].Q_0
U 11EC. 0D00.003C.0180.0800.0000.11ED  ;2488  D_DAL.SC
U 11ED. 0000.003C.8D80.0800.0084.71EE  ;2489  SC_K[.1F]
U 11EE. 0801.001C.0180.0800.0000.11EF  ;2490  D_D.ORNOT.MASK
U 11EF. 0000.003E.0180.0800.0000.0001  ;2491  RETURN1    ;D=880F0000
            ;2492
```

```
            :2493  .PAGE
            :2494  .TOC "   GET SIL COMP REG DATA IN D FOR ID COMPARE LOCK"
            :2495 ;
            :2496 ;*******************************************************************************
            :2497 ;**
            :2498 ;
            :2499 ; THIS ROUTINE PUTS THE DATA FOR THE SILO COMPARATER
            :2500 ; REGISTER TO LOCK ON ID COMPARE ONLY AND AN E IN THE
            :2501 ; COUNT FIELD, INTO THE D REGISTER.
            :2502 ;
            :2503 ;--
            :2504 ;*******************************************************************************
            :2505 ;
U 11F0. 0818.0038.4180.0800.0000.104A  :2506 STCOMP: D_K[.80]
U 104A. 0000.003D.0180.0800.0000.11E4  :2507 =0 CALL,J/DCE    ;SET LOCK ON ID COMPARE
U 104B. 0000.003C.65F8.0800.0084.71F1  :2508  SC_K[.10],Q_0    ;AND E IN THE COUNT FIELD
U 11F1. 0D00.003E.0180.0800.0000.0001  :2509  D_DAL.SC.RETURN1    ;D=080E0000
            :2510
```

```
            :2511  .PAGE
            :2512  .TOC "   SET MAINT ID AND TURN OFF CACHE"
            :2513 ;
            :2514 ;*****************************************************************************
            :2515 ;**
            :2516 ;
            :2517 ; THIS ROUTINE SETS UP THE MAINTENANCE REGISTER
            :2518 ; DATA TO DISABLE CACHE AND RC0 <4:0> ARE PUT
            :2519 ; IN BITS <27:23> (MAINTENANCE ID). THE DATA
            :2520 ; IS RETURNED IN THE D REGISTER. WHEN THIS ROUTINE
            :2521 ; IS CALLED RC0 <4:0> MUST CONTAIN THE ID TO BE
            :2522 ; USED IN THE MAINTENANCE ID.
            :2523 ;
            :2524 ;--
            :2525 ;*****************************************************************************
            :2526 ;
U 11F2. 0810.0038.0180.0900.0000.11F3  ;2527 SETMRG: D_RC[0]
U 11F3. 0018.0038.7DC0.0800.0000.11F4  ;2528  Q_K[.18]
U 11F4. 0019.2000.09F8.0800.0082.11F5  ;2529  SC_Q-K[.2],Q_0
U 11F5. 0D00.003C.0180.0800.0000.11F6  ;2530  D_DAL.SC    ;SHIFT ID INTO BITS <27:23>
U 11F6. 0819.0030.4580.0800.0000.11F7  ;2531  D_D.OR.K[.8000]
U 11F7. 0500.003C.0180.0800.0000.11F8  ;2532  D_D.LEFT    ;DISABLE CACHE
U 11F8. 0819.0032.4580.0800.0000.0001  ;2533  D_D.OR.K[.8000],RETURN1
            :2534
```

```
            ;2535  .PAGE
            ;2536  .TOC "   SET SIL COMP REG TO LOCK ON ID AND TAG COMPARE"
            ;2537 ;
            ;2538 ;*******************************************************************************
            ;2539 ;**
            ;2540 ;
            ;2541 ; THIS ROUTINE SETS UP THE SILO COMPARATOR REGISTER
            ;2542 ; CONDITIONAL LOCK BITS <28:27> TO LOCK ON AN ID AND
            ;2543 ; TAG COMPARISON AND PUTS AN E IN THE COUNT FIELD.
            ;2544 ; THE TEST TAG MUST BE IN RC C WHEN THIS ROUTINE IS
            ;2545 ; CALLED. THE DATA IS RETURNED IN D.
            ;2546 ;
            ;2547 ;--
            ;2548 ;*******************************************************************************
            ;2549 ;
U 11F9. 0810.0038.0180.0960.0000.104C  ;2550 TGCOMP: D_RC[0C]    ;TEST TAG INTO D
U 104C. 0000.003D.0180.0800.0000.11E4  ;2551 =0 CALL,J/DCE    ;E FOR COUNT FIELD
U 104D. 0000.003C.65F8.0800.0084.71FA  ;2552  SC_K[.10],Q_0    ;SHIFT INTO BITS <22:16>
U 11FA. 0D00.003C.0180.0800.0000.11FB  ;2553  D_DAL.SC
U 11FB. 0000.003C.5180.0800.0084.71FC  ;2554  SC_K[.1E]
U 11FC. 0000.003C.0980.0800.0084.B1FD  ;2555  SC_SC-K[.2]
U 11FD. 0801.001C.0180.0800.0000.11FE  ;2556  D_D.ORNOT.MASK    ;SET <28> LOCK ON ID AND TAG
U 11FE. 0000.003E.0180.0800.0000.0001  ;2557  RETURN1
            ;2558
```

```
            ;2559  .PAGE
            ;2560  .TOC "   SET SIL COMP REG TO LOCK ON ID, TAG AND FUNC"
            ;2561 ;
            ;2562 ;*****************************************************************************
            ;2563 ;**
            ;2564 ;
            ;2565 ; THIS ROUTINE SETS UP THE SILO COMPARATOR REGISTER TO LOCK ON
            ;2566 ; AN ID,TAG AND FUNCTION COMPARISON. THE TAG FIELD <22:20> IS
            ;2567 ; SET TO A COMMAND ADDRESS TAG OF 3. THE COUNT FIELD IS SET TO E.
            ;2568 ; THE FUNCTION WHICH IS TO BE COMPARED MUST BE IN RCA <3:0> WHEN
            ;2569 ; THIS ROUTINE IS CALLED. THE DATA IS RETURNED IN THE D REGISTER.
            ;2570 ;
            ;2571 ;--
            ;2572 ;*****************************************************************************
            ;2573 ;
U 11FF, 0810.0038.0180.0950.0000.1200  ;2574 FNCOMP: D_RC[0A]
U 1200, 0819.0030.7980.0800.0000.1050  ;2575  D_D.OR.K[.30]
U 1050, 0000.003D.0180.0800.0000.11DC  ;2576 =0 CALL,J/DC6
U 1051, 0600.003C.0180.0800.0000.1052  ;2577  D_D.RIGHT
U 1052, 0000.003D.0180.0800.0000.11E4  ;2578 =0 CALL,J/DCE
U 1053, 0000.003C.65F8.0800.0084.7201  ;2579  SC_K[.10],Q_0
U 1201, 0D00.003C.0180.0800.0000.1202  ;2580  D_DAL.SC
U 1202, 0000.003E.0180.0800.0000.0001  ;2581  RETURN1
            ;2582
```

```
;2583  .PAGE
;2584  .TOC "   FIND MS780-E MEMORY"
;2585 ;**
;2586 ; FUNCTIONAL DESCRIPTION:
;2587 ;
;2588 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
;2589 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
;2590 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
;2591 ;     3D will hold the I/O base address of the MS780-E subsystems found
;2592 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
;2593 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
;2594 ;     first MS780-E found will have  its  starting  address  set  to  0
;2595 ;     (zero).
;2596 ;     All the other MS780-E/H's found will have their starting  address
;2597 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
;2598 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
;2599 ;     apropriate  action.  Register RC[0E] is used as a pointer to the
;2600 ;     current MS780-E unit under test.
;2601 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
;2602 ;     set to maximum.
;2603 ;
;2604 ; CALLING CONSTRAINT:
;2605 ;
;2606 ; =00
;2607 ;
;2608 ; OUTPUTS:
;2609 ;
;2610 ; rc[0e] - Contains address of Configuration Register
;2611 ; r[0]  - Contains TR level
;2612 ;
;2613 ; SIDE EFFECTS:
;2614 ;
;2615 ; D register is destroyed.
;2616 ;--
;2617 =0
;2618 FIND.MS780E:
U 1054. 0000,003D,0180,0800,0000,102E  ;2619  call[sbi.init]  ; Make sure SBI is enabled
U 1055. 0003,003C,0180,0988,0000,1203  ;2620  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1203. 0818,0038,1980,0800,0000,1204  ;2621  d_k[zero]  ; Clear MS780-E/H counter
U 1204. 0000,003C,F980,3C00,0000,1205  ;2622  id[3e]_d  ; ...
U 1205. 0018,0038,0580,0A80,0000,1206  ;2623  r[0]_k[.1]  ; Init TR counter
U 1206. 0F03,003C,0180,09F0,0000,1056  ;2624  d_0,rc[0E]_0  ; Clear "Save" location for
;2625     ; ...CNFG A Reg
;2626 =0
;2627 FIND.MEM.LOOP:
U 1056. 0800,003D,0180,0A00,0000,122C  ;2628  d_r[0],call[generate.io]; Generate address of TR level
U 1057. 0001,003C,0180,09F0,0200,1058  ;2629  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1058. 0000,003D,8980,0800,0084,7199  ;2630 =0 set.trap.mask[.d] ; Enable timeout micro traps
;2631  d[long]_cache.p, ; Read the specified tr level
U 1059. 0000,003C,0180,C970,0000,1207  ;2632   ic_rc[0e]  ; ...
U 1207. 0000,003C,0180,0A78,0010,1208  ;2633  alu_r[0f],clk.ubcc ; Check for no response
U 1208. 0001,013C,0180,0880,0082,105C  ;2634  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 105C. 0000,003C,0180,0800,0000,1209  ;2635 =0 j/check.adpt.code ; Check for a memory
U 105D. 0000,003C,0180,0800,0000,1228  ;2636  j/find.mem.lp1  ; Try next tr
;2637 CHECK.ADPT.CODE:
```

```
U 1209, 0000.003C.1D80.0800.1404.720A  ;2638  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 120A, 0000.163C.0180.0800.0000.1008  ;2639  state7-4?  ; Branch on the adapter type
U 1008, 0000.003C.8980.0800.0084.720B  ;2640 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000.003C.8980.0800.0084.720C  ;2641  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000.003C.0180.0800.0000.1228  ;2642  j/find.mem.lp1  ; Not a memory
U 100B, 0000.003C.0180.0800.0000.1228  ;2643  j/find.mem.lp1  ; Not a memory
U 100C, 0000.003C.6580.0800.0084.7211  ;2644  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000.003C.0180.0800.0000.1228  ;2645  j/find.mem.lp1  ; Not a memory
U 100E, 0010.0038.0180.09F0.0000.1215  ;2646  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010.0038.0180.09F0.0000.1215  ;2647  rc[0e]_lc,j/found.ms780e ; MS780-E
           ;2648   ;
           ;2649   ; Found an MS780-A or -C. Set the starting address
           ;2650   ; to maximum.
           ;2651   ;
U 120B, 0818.0038.5D80.0800.0000.120D  ;2652 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 120C, 0818.0038.0580.0800.0000.120D  ;2653 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
           ;2654 MS780.COMMON:
U 120D, 0819.0030.7180.0800.0000.120E  ;2655  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 120E, 0000.003C.01F8.0803.0080.D20F  ;2656  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 120F, 0D00.003C.0180.0800.0000.1210  ;2657  d_dal.sc  ; Put data in bits<29:14>
           ;2658  cache.p_d[long], ; Set the starting address to maximum
U 1210, 0000.003C.0180.A800.0000.1228  ;2659       j/find.mem.lp1 ; and continue TR scan
           ;2660   ;
           ;2661   ; Found an MA780. Set the starting address to maximum
           ;2662   ;
           ;2663 MA780.MAX:
U 1211, 0818.0038.39F8.0803.0000.1212  ;2664  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1212, 0D00.003C.0180.0803.0000.1213  ;2665  d_dal.sc,va_va+4 ; Put in proper position
U 1213, 0000.003C.0180.0803.0000.1214  ;2666  va_va+4   ; Point to Port Invalidate Ctrl Register
           ;2667  cache.p_d[long], ; Set starting address to maximum
U 1214, 0000.003C.0180.A800.0000.1228  ;2668       j/find.mem.lp1
           ;2669 ;
           ;2670 ; Found an MS780E
           ;2671 ;
           ;2672 FOUND.MS780E:
U 1215, 0000.003C.F9F0.2C00.0000.1216  ;2673  q_id[3e]  ; Set up to see how many MS780-E's
           ;2674  alu_q.xor.k[.3], ; Are there Maximum units?
U 1216, 0019.2020.0D80.0800.0010.1217  ;2675   clk.ubcc ; Check condition codes
U 1217, 0000.013C.0180.0800.0000.1060  ;2676  z?   ; Are we at maximum units for system
U 1060, 0000.003C.0180.0800.0000.1218  ;2677 =0 J/ms780e.tst  ; No go find how many units ther are
U 1061, 0818.0038.C180.0800.0000.1062  ;2678  d_k[.ffff]  ; Yes set address to maximum
U 1062, 0000.003D.0180.0800.0000.1230  ;2679 =0 call[set.start.addr] ; .....
U 1063, 0000.003C.0180.0800.0000.1228  ;2680  j/find.mem.lp1  ; Go check to see if we are done
           ;2681 ;
           ;2682 MS780E.TST:
           ;2683  alu_rc[01],  ; Check "Found MS780E Flag"
U 1218, 0010.0038.0180.0908.0010.1219  ;2684   clk.ubcc ; Check condition codes
U 1219, 0810.0138.0180.0970.0000.1064  ;2685  z?,d_rc[0e]  ; Is this unit the first MS780E ?
           ;2686 =0 j/not.first.ms780e, ; No
U 1064, 0818.0038.C180.0800.0000.1068  ;2687   d_k[.ffff] ; Set starting address
U 1065, 0001.003C.0180.0988.0000.121A  ;2688  rc[01]_d  ; Yes put base address in rc[01]
U 121A, 0818.0038.7980.0800.0000.121B  ;2689  d_k[.30]  ; Set up SC to point to first unit
U 121B, 0019.0014.F580.0800.0082.121C  ;2690  alu_d+k[.a],sc_alu ; ...
U 121C, 0810.0038.0180.0908.0000.121D  ;2691  d_rc[01]  ; Put base address of unit in D
U 121D, 0000.003C.0180.3400.0000.121E  ;2692  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 121E. 0F00.003C.0180.0800.0000.1066  ;2693  d_0   ; Prepare to set starting address to Zero
U 1066. 0000.003D.0180.0800.0000.1230  ;2694 =0 call[set.start.addr] ; Go do it
        ;2695  j/find.mem.lp1,  ; Check to see if we are done
U 1067. 0003.003C.0180.09E8.0000.1228  ;2696   rc[0d]_0 ; ....
        ;2697 =0
        ;2698 NOT.FIRST.MS780E:
U 1068. 0000.003D.0180.0800.0000.1230  ;2699  call[set.start.addr] ; Go set starting address to maximum
U 1069. 0000.003C.F9F0.2C00.0000.121F  ;2700  q_id[3e]  ; Get the number of MS780-Es found
U 121F. 0819.2014.0580.0800.0000.1220  ;2701  d_q+k[.1]  ; Increment this counter
U 1220. 0000.003C.F980.3C00.0000.1221  ;2702  id[3e]_d  ; Save the counter
U 1221. 0018.0038.11C0.0800.0000.1222  ;2703  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2704     ; ...were the I/O base addresses will be stored
U 1222. 081D.2000.7980.0800.0000.1223  ;2705  d_q-d,k[.30]  ; ... adjust pointer
U 1223. 0819.0014.7980.0800.0000.1224  ;2706  d_d+k[.30]  ; Point the SC to the ID register
U 1224. 0000.003C.F580.0800.0000.1225  ;2707  k[.a]   ; ...to receive I/O base address
U 1225. 0019.0014.F580.0800.0082.1226  ;2708  alu_d+k[.a],sc_alu ; ...
U 1226. 0810.0038.0180.0970.0000.1227  ;2709  d_rc[0e]  ; Get base address to d
U 1227. 0000.003C.0180.3400.0000.1228  ;2710  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2711   ;
        ;2712   ; Try next tr
        ;2713   ;
        ;2714 FIND.MEM.LP1:
U 1228. 0818.0014.0580.0A80.0000.1229  ;2715  r[0]_la+k[.1],d_alu ; Increment TR level
        ;2716  alu_d.and.k[.f],
U 1229. 0019.0034.6180.0800.0010.122A  ;2717      clk.ubcc  ; Check if all done
U 122A. 0000.013C.0180.0800.0000.106A  ;2718  z?   ; Branch if yes
U 106A. 0000.003C.0180.0800.0000.1056  ;2719 =0 j/find.mem.loop  ; Try next TR
U 106B. 0810.0038.0180.0908.0010.122B  ;2720  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 122B. 0000.013C.0180.0800.0000.106C  ;2721  z?   ; Check condition codes
U 106C. 0001.003E.0180.09F0.0000.0002  ;2722 =0 return2,rc[0e]_d ; Yes MS780E was found
U 106D. 0000.003E.0180.0800.0000.0001  ;2723  return1   ; No MS780E found
        ;2724
```

```
            ;2725  .PAGE
            ;2726  .TOC "   Generate IO Address"
            ;2727 ;**
            ;2728 ; FUNCTIONAL DESCRIPTION:
            ;2729 ;
            ;2730 ; This routine is used to generate an SBI Configuration Register
            ;2731 ; address from a TR level.
            ;2732 ;
            ;2733 ; CALLING CONSTRAINT:
            ;2734 ;
            ;2735 ; =0
            ;2736 ;
            ;2737 ; INPUTS:
            ;2738 ;
            ;2739 ; D - Contains TR level
            ;2740 ;
            ;2741 ; OUTPUTS:
            ;2742 ;
            ;2743 ; D - Contains SBI IO address
            ;2744 ;--
            ;2745 GENERATE.IO:
U 122C, 0000,003C,89F8,0800,0084,722D  ;2746  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 122D, 0D00,003C,8D80,0800,0084,722E  ;2747  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 122E, 0000,003C,0980,0800,0084,B22F  ;2748  sc_sc-k[.2]  ; insert bit 29
U 122F, 0801,001E,0180,0800,0000,0001  ;2749  d_d.ornot.mask,return1 ; and return
            ;2750
```

```
            ;2751  .PAGE
            ;2752  .TOC "   Set Starting Address"
            ;2753 ;++
            ;2754 ; FUNCTIONAL DESCRIPTION:
            ;2755 ;
            ;2756 ; This routine is used to set the starting address of the
            ;2757 ; memory to the specified value.
            ;2758 ;
            ;2759 ; CALLING CONSTRAINT:
            ;2760 ;
            ;2761 ; =0
            ;2762 ;
            ;2763 ; INPUTS:
            ;2764 ;
            ;2765 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2766 ;     (B Register Image divided by 2**16)
            ;2767 ; rc[0] - Contains Address of Register A
            ;2768 ;
            ;2769 ; OUTPUTS:
            ;2770 ;
            ;2771 ; d - Contains aligned starting address (B Register Image)
            ;2772 ;
            ;2773 ; SIDE EFFECTS:
            ;2774 ;
            ;2775 ; d,q,va, and sc are destroyed
            ;2776 ;--
            ;2777 SET.START.ADDR:
U 1230, 0010,0038,6580,0970,0284,7231  ;2778  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1231, 0000,003C,01F8,0803,0000,1232  ;2779  va_va+4,q_0  ; Set address to Register B
            ;2780  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1232, 0D00,003C,0980,0800,0084,B233  ;2781       sc_sc-k[.2]  ; Setup to insert bit 14
U 1233, 0801,001C,0180,0800,0000,1234  ;2782  d_d.ornot.mask  ; Insert Write Enable bit
U 1234, 0000,003E,0180,A800,0000,0001  ;2783  cache.p_d[long],return1 ; Set the starting address and return
            ;2784
```

```
            ;2785  .PAGE
            ;2786  .TOC "   Select Controller"
            ;2787 ;**
            ;2788 ; FUNCTIONAL DESCRIPTION:
            ;2789 ;
            ;2790 ; This routine is used to put the memory system into the
            ;2791 ; specified configuration with respect to controller select.
            ;2792 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2793 ; a RETURN2 is made.
            ;2794 ;
            ;2795 ; CALLING CONSTRAINT:
            ;2796 ;
            ;2797 ; =00
            ;2798 ;
            ;2799 ; INPUTS:
            ;2800 ;
            ;2801 ; d - Contains value to write to bits<2:0> of Register A
            ;2802 ; rc[0e] - Address of Register A
            ;2803 ;
            ;2804 ; SIDE EFFECTS:
            ;2805 ;
            ;2806 ; sc,d,va are destroyed
            ;2807 ;--
            ;2808 SELECT.CNTRLR:
U 1235, 0010,003B,0180,0970,0284,7236  ;2809  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1236, 0801,001C,0180,0800,0000,1237  ;2810  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1237, 0000,003C,0180,A800,0000,1238  ;2811  cache.p_d[long]  ; Write the configuration Register
U 1238, 0818,003B,5D80,0800,0000,1239  ;2812  d_k[.7]   ; Get Misconfiguration bits
U 1239, 0000,003C,61F8,0800,0084,723A  ;2813  sc_k[.F],q_0  ; ...
U 123A, 0D00,003C,0180,0800,0000,123B  ;2814  d_dal.sc  ; ... D = x38000
U 123B, 0000,003C,01E0,0800,0000,123C  ;2815  q_d   ; Save mask
U 123C, 0000,003C,0180,C800,0000,123D  ;2816  d[long]_cache.p  ; Read CNFG A Reg and
            ;2817  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 123D, 001D,2034,0180,0800,0010,123E  ;2818   clk.ubcc ; ...
U 123E, 0000,013C,0180,0800,0000,106E  ;2819  z?   ; Branch if no
U 106E, 0000,003E,0180,0800,0000,0001  ;2820 =0 RETURN1   ; Bad configuration
U 106F, 0000,003E,0180,0800,0000,0002  ;2821  RETURN2   ; Configuration ok
            ;2822
```

```
;2823  .PAGE
;2824  .TOC "   SELECT LOWER CONTROLLER"
;2825 ;*************************************************************
;2826 ;++
;2827 ;
;2828 ; Functional Description:
;2829 ; ----------------------
;2830 ;
;2831 ; This subroutine can be called to select the lower
;2832 ; Controller on a MS780-E/H.
;2833 ; If the Lower controller is misconfigured then a
;2834 ; RETURN1 will be made.  Otherwise a RETURN2
;2835 ;
;2836 ; Calling Constraint: =00
;2837 ; ------------------
;2838 ;
;2839 ;
;2840 ; Inputs:
;2841 ; ------
;2842 ;
;2843 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2844 ;
;2845 ; Outputs:
;2846 ; -------
;2847 ;
;2848 ; RC[0D] will have the address of CNFG Register C
;2849 ;        ( 2000x008 )
;2850 ;
;2851 ; Side Effects:
;2852 ; ------------
;2853 ;
;2854 ; Contents of the following registers will lost:
;2855 ;
;2856 ; D
;2857 ;
;2858 ; The Lower controller on the MS780-E/H will be configured
;2859 ; when the subroutine is exitted with a RETURN2
;2860 ;--
;2861 ;*************************************************************
;2862 =00
;2863 SELECT.LOWER:
;2864  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,1235  ;2865   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2866  RETURN1    ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2867  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2868  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2869   RETURN2   ; Let caller know that the controller
;2870       ; ...was configured properly
;2871
```

```
;2872  .PAGE
;2873  .TOC "   SELECT UPPER CONTROLLER"
;2874 ;*************************************************************
;2875 ;**
;2876 ;
;2877 ; Functional Description:
;2878 ; ----------------------
;2879 ;
;2880 ; This subroutine can be called to select the upper
;2881 ; Controller on a MS780-E/H.
;2882 ; If the Upper controller is misconfigured then a
;2883 ; RETURN1 will be made.  Otherwise a RETURN2
;2884 ;
;2885 ; Calling Constraint: =00
;2886 ; ------------------
;2887 ;
;2888 ;
;2889 ; Inputs:
;2890 ; ------
;2891 ;
;2892 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2893 ;
;2894 ; Outputs:
;2895 ; -------
;2896 ;
;2897 ; RC[0D] will have the address of CNFG Register D
;2898 ;        ( 2000x010 )
;2899 ;
;2900 ; Side Effects:
;2901 ; ------------
;2902 ;
;2903 ; Contents of the following registers will lost:
;2904 ;
;2905 ; D
;2906 ;
;2907 ; The Upper controller on the MS780-E/H will be configured
;2908 ; when the subroutine is exitted with a RETURN2
;2909 ;--
;2910 ;*************************************************************
;2911 =00
;2912 SELECT.UPPER:
;2913  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,1235  ;2914   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2915  RETURN1    ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2916  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2917  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2918   RETURN2   ; Let caller know that the controller
;2919       ; ...was configured properly
```

```
            ;2920  .PAGE
            ;2921  .TOC "   ENABLE NEXT MS780-E"
            ;2922 ;++
            ;2923 ; FUNCTIONAL DESCRIPTION:
            ;2924 ;
            ;2925 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2926 ;    They will be tested  separately, according to the TR level at which
            ;2927 ;    they are located,  starting with the one  at the lowest level first.
            ;2928 ;          When  a test has finished  with the first unit, it will have to call
            ;2929 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2930 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2931 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2932 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2933 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2934 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2935 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2936 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2937 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2938 ;
            ;2939 ; CALLING CONSTRAINT:
            ;2940 ;
            ;2941 ;  =00
            ;2942 ;
            ;2943 ;--
            ;2944 ENABLE.NEXT.MS780E:
U 123F, 0000.003C.F9F0.2C00.0000.1240  ;2945  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2946  ALU_Q,   ; Check to see if it is zero
U 1240, 0001.203C.0180.0800.0010.1241  ;2947     CLK.UBCC  ; Check Condition Codes
U 1241, 0000.013C.0180.0800.0000.1070  ;2948  Z?   ; ...for zero
U 1070, 0000.003C.0180.0800.0000.1242  ;2949 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1071, 0000.003E.0180.0800.0000.0002  ;2950  RETURN2   ; Zero No more units to test
            ;2951 ENABLE.NEXT:
U 1242, 0818.0038.C180.0800.0000.1072  ;2952  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1072, 0000.003D.0180.0800.0000.1230  ;2953 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1073, 0818.0038.7980.0800.0000.1243  ;2954  D_K[.30]  ; Prepare to point to the ID register holding
            ;2955     ; ...the I/O Base address of the next MS780-E
U 1243, 0819.0014.1180.0800.0000.1244  ;2956  D_D+K[.4]  ; ...
U 1244, 0000.003C.F580.0800.0000.1245  ;2957  K[.A]   ; ...
U 1245, 0819.0014.F580.0800.0000.1246  ;2958  D_D+K[.A]  ; ...
U 1246, 0000.003C.F9F0.2C00.0000.1247  ;2959  Q_ID[3E]  ; Get MS780-E Counter
            ;2960  D_D-Q,   ; D now holds the pointer to the ID register
U 1247, 081D.0000.0180.0800.0082.1248  ;2961    SC_ALU  ; ...
U 1248, 0000.003C.01F0.2400.0000.1249  ;2962  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1249, 0001.203C.0180.09F0.0000.124A  ;2963  RC[0E]_Q  ; Save it also in RC[0E]
U 124A, 0F03.003C.0180.09E8.0000.1074  ;2964  RC[0D]_0,D_0  ; Set starting address to zero
U 1074, 0000.003D.0180.0800.0000.1230  ;2965 =0 CALL[SET.START.ADDR] ; ....
U 1075, 0F00.003C.F9F0.2C00.0000.124B  ;2966  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 124B, 081D.2008.0180.0800.0000.124C  ;2967  D_Q-D-1   ; Subtract 1 from total
U 124C, 0000.003C.F980.3C00.0000.1076  ;2968  ID[3E]_D  ; Adjust the pointer
U 1076, 0000.003D.0180.0800.0000.119C  ;2969 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 1077, 0000.003E.0180.0800.0000.0001  ;2970  RETURN1   ; Tell caller another unit was found
            ;2971
```

```
            ;2972  .PAGE
            ;2973  .TOC "   LOAD TEMPORARY REGISTERS"
            ;2974 ;**
            ;2975 ; FUNCTIONAL DESCRIPTION
            ;2976 ;
            ;2977 ; This subroutine will be used by tests that check the data
            ;2978 ; integrity of some registers or memory location. It loads
            ;2979 ; the ID Temporary registers, T1 through T7 with some data
            ;2980 ; found in cache, the address of which has been previously
            ;2981 ; set up (before the call is made) by the calling routine
            ;2982 ; in the VA register.
            ;2983 ;
            ;2984 ; CALLING CONSTRAINT:
            ;2985 ;
            ;2986 ; =0
            ;2987 ;
            ;2988 ;
            ;2989 ; SIDE EFFECTS:
            ;2990 ;
            ;2991 ; The SC register will be used in this case as a pointer to
            ;2992 ; the ID Temporary registers.
            ;2993 ;
            ;2994 ;--
            ;2995 LOAD.TEMP.REGS:
U 124D, 0018,0038,5D80,0A88,0000,1078  ;2996  R[1]_K[.7]  ; Set up counter for 7 temporary registers
U 1078, 0000,003D,0180,0800,0000,1191  ;2997 =0 CALL[ENABLE.CACHE] ; Enable the cache
U 1079, 001B,0010,7980,0800,0082,124E  ;2998  SC_K[.30]+1  ; Use the SC register as a pointer to the
            ;2999      ; ... ID temporary registers
            ;3000 LOAD.TEMP.LOOP1:
U 124E, 0000,003C,0180,C800,0000,124F  ;3001  D[LONG]_CACHE.P  ; Read a longword from cache
U 124F, 0000,003C,0180,3400,0000,1250  ;3002  ID(SC)_D  ; Leave the data in Temporary
            ;3003      ; ...register pointed by the SC
U 1250, 0000,003C,0180,0800,0080,D251  ;3004  SC_SC+1   ; Point to next Temp.REG
U 1251, 0000,003C,0180,0803,0000,1252  ;3005  VA_VA+4   ; Point to next longword of data in cache
U 1252, 0800,003C,0180,0A08,0000,1253  ;3006  D_R[1]    ; Prepare to decrement the counter
U 1253, 0019,0000,0580,0A88,0010,1254  ;3007  R[1]_D-K[.1],CLK.UBCC ; Decrement the counter by 1
U 1254, 0000,013C,0180,0800,0000,107A  ;3008  Z?    ; Is the counter zero ?
U 107A, 0000,003C,0180,0800,0000,124E  ;3009 =0 J/LOAD.TEMP.LOOP1 ; Jump if not zero else..
U 107B, 0000,003C,0180,0800,0000,107C  ;3010  NOP
U 107C, 0000,003D,0180,0800,0000,1195  ;3011 =0 CALL[DISABLE.CACHE] ; Disable the cache
U 107D, 0000,003E,0180,0800,0000,0001  ;3012  RETURN1   ; Exit
            ;3013
```

```
;3014  .PAGE
;3015  .TOC "   CONFIGURE MS780-E/H"
;3016 ;*************************************************************
;3017 ;**
;3018 ;
;3019 ; Functional Description:
;3020 ; ----------------------
;3021 ;
;3022 ;      This  subroutine  will  be  used  by  tests  that   do   not
;3023 ; specifically  check  the  memory, but  make use of it to execute.
;3024 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;3025 ; that  if  a  Read/Write operation will take place no false errors
;3026 ; will be logged due to misconfigured memory.
;3027 ;
;3028 ; Calling Constraint: =00
;3029 ; ------------------
;3030 ;
;3031 ;
;3032 ; Inputs:
;3033 ; ------
;3034 ;
;3035 ; RC[0E] - I/O BASE OF MS780-E/H
;3036 ;
;3037 ;
;3038 ; Outputs:
;3039 ; -------
;3040 ;
;3041 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;3042 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;3043 ;      OR IF NO MS780-E/Hs WERE FOUND
;3044 ;      ON THE SBI
;3045 ;
;3046 ; Side Effects:
;3047 ; ------------
;3048 ;
;3049 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;3050 ;
;3051 ;
;3052 ;
;3053 ;--
;3054 ;*************************************************************
;3055 CONFIG.MS780E:
U 1255, 0818,0038,0D80,0800,0000,1256  ;3056  D_K[.3]    ; Set up flag for the Fail Chain
U 1256, 0001,003C,0180,09E8,0000,1014  ;3057  RC[0D]_D    ; ...
U 1014, 0000,003D,0180,0800,0000,1054  ;3058 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1015, 0000,003D,0980,0800,0084,702A  ;3059  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 1016, 0000,003C,0180,0800,0000,1018  ;3060  NOP     ; OK. Found at least one MS780-E/H
;3061 =
;3062 CONFIG.RETRY:     ; Entry point for the case in which the
;3063       ; ...first MS780-E/H could not be
;3064       ; ...properly configured
U 1018, 0000,003D,0180,0800,0000,1004  ;3065 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1019, 0000,003C,0180,0800,0000,101C  ;3066  J/CNFG.LMIS    ; Lower controller misconfigured
U 101A, 0000,003E,0180,0800,0000,0001  ;3067  RETURN1     ; OK. Lower Controller is configured
;3068 =
```

```
            ;3069 =00
            ;3070 CNFG.LMIS:
U 101C. 0000.003D.0180.0800.0000.1010  ;3071  CALL[SELECT.UPPER]  ; Select the upper controller
U 101D. 0000.003C.0180.0800.0000.1020  ;3072  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 101E. 0000.003E.0180.0800.0000.0001  ;3073  RETURN1    ; OK. Upper Controller is configured
            ;3074 =
            ;3075 =00
            ;3076 CNFG.UMIS:
U 1020. 0000.003D.0180.0800.0000.123F  ;3077  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;3078       ; ...MS780-E/H found so go and see
            ;3079       ; ...if there are any more
U 1021. 0000.003C.0180.0800.0000.1257  ;3080  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;3081       ; ...try to configure it
U 1022. 0818.0038.0D80.0800.0000.1023  ;3082  D_K[.3]    ; Set up flag for the Fail Chain
U 1023. 0001.003C.0180.09E8.0000.1257  ;3083  RC[0D]_D    ; ...
U 1257. 0000.003D.0980.0800.0084.702A  ;3084  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;3085       ; ...abort the test
            ;3086 =
            ;3087
```

```
;3088  .PAGE "TEST 191 DATA INTEGRITY OF ID REG 1C"
;3089 ;*************************************************************
;3090 ;***
;3091 ;
;3092 ; Functional Description:
;3093 ; ----------------------
;3094 ;
;3095 ;      This  test  checks  out  the  integrity  of  the  SBI   Silo
;3096 ; Comparator Register (ID reg.  1C) by doing consecutive Write/Read
;3097 ; operations with the data patterns detailed in Appendix  B.  These
;3098 ; test patterns will be stored in cache and they are as follows:
;3099 ;
;3100 ;
;3101 ; Mask : x7FFF0000
;3102 ;
;3103 ;  x2AAA0000
;3104 ;
;3105 ;  x55550000
;3106 ;
;3107 ;  x33330000
;3108 ;
;3109 ;  x0F0F0000
;3110 ;
;3111 ;  x7F000000
;3112 ;
;3113 ;  x7FFF0000
;3114 ;
;3115 ;
;3116 ; CACHE DATA FOR THIS TEST:
;3117 ;
;3118  ;% 8
;3119  ;$ 0000,7FFF,0000,2AAA,0000,5555,0000,3333
;3120  ;$ 0000,0F0F,0000,7F00,0000,7FFF
;3121 ;
;3122 ; Logic Description:
;3123 ; ------------------
;3124 ;  N/A
;3125 ;
;3126 ; Error Logout:
;3127 ; ------------
;3128 ;
;3129 ; See individual error reports.
;3130 ;
;3131 ; Sync Point Description:
;3132 ; ----------------------
;3133 ;
;3134 ;   [If necessary]
;3135 ;
;3136 ;=============================================================
;3137 ;
;3138 ; Register Map:
;3139 ; ------------
;3140 ;
;3141 ;      RA,RB      Description:
;3142 ;      -----      -----------
```

```
            ;3143 ;
            ;3144 ;       0       LOOP COUNTER
            ;3145 ;
            ;3146 ;       1       POINTER TO TESTS PATTERNS IN TEMP REGS
            ;3147 ;
            ;3148 ;
            ;3149 ;      RC       Description:
            ;3150 ;      --       -----------
            ;3151 ;
            ;3152 ; C BIT MASK
            ;3153 ;
            ;3154 ; D RECEIVED DATA FROM ID REG 1C
            ;3155 ;
            ;3156 ;       E       EXPECTED DATA FROM ID REG 1C
            ;3157 ;
            ;3158 ;
            ;3159 ;--
            ;3160 ;*****************************************************************
            ;3161 =0
U 107E, 0000,003D,0180,0800,0000,117C  ;3162 TST.9: NEWTST    ; Increment test number
U 107F, 0018,0038,0180,0800,0200,1080  ;3163  VA_K[.8]    ; Point to data in cache
U 1080, 0000,003D,0180,0800,0000,124D  ;3164 =0 CALL[LOAD.TEMP.REGS]  ; Load test data from cache in
            ;3165       ; ...Temporary Registers 1
            ;3166       ; ...through 7
U 1081, 0818,0038,2980,0800,0000,1258  ;3167  D_K[.34]    ; Form x31 in the SC
            ;3168  SC_D-K[.3],    ; ...
U 1258, 0019,0000,0D80,0A88,0082,1259  ;3169   R[1]_ALU  ; Save in R[1]
U 1259, 0000,003C,01F0,2400,0000,125A  ;3170  Q_ID(SC)    ; Get bit Mask from Temporary
            ;3171       ; ...Register 1
U 125A, 0001,203C,0180,09E0,0000,125B  ;3172  RC[0C]_Q    ; Save in RC[0C]
U 125B, 0000,003C,0180,0888,0000,125C  ;3173  LA_RA[1]
U 125C, 001B,0014,0580,0A88,0000,125D  ;3174  R[1]_LA+K[.1]    ; Point to first Test Pattern in
            ;3175       ; ...Temporary Register 2
U 125D, 0018,0038,D580,0A80,0000,1082  ;3176  R[00]_K[.6]    ; Set up a loop counter
            ;3177 =0
            ;3178 TEST.9.LOOP1:
U 1082, 0000,003D,0180,0800,0000,119D  ;3179  ERLOOP    ; Set the error looping mechanism
U 1083, 0000,003C,0180,0A08,0082,125E  ;3180  SC_R[1]    ; Point to Test Pattern
U 125E, 0000,003C,01F0,2400,0000,125F  ;3181  Q_ID(SC)    ; Get a Test Pattern
            ;3182  RC[0E]_Q,    ; Save it in RC[0E] for error logout
U 125F, 0801,203C,0180,09F0,0000,1260  ;3183   D_ALU    ; Get it to the D Reg
U 1260, 0000,003C,7180,3C00,0000,1261  ;3184  ID[COMP]_D    ; Write the Test Pattern to ID COMP Reg
            ;3185  Q_ID[COMP],    ; Read contents of ID COMP Reg
U 1261, 0000,003C,71F0,2D60,0000,1262  ;3186   LC_RC[0C]  ; Get Bit Mask
            ;3187  D_Q.AND.LC,    ; Mask unwanted bits
U 1262, 0811,2034,0180,09E8,0000,1263  ;3188   RC[0D]_ALU  ; Save for error logout
            ;3189  ALU_D.XOR.RC[0E],  ; Is the received data equal to the
U 1263, 0011,0020,0180,0970,0010,1264  ;3190   CLK.UBCC  ; ...expected
            ;3191  Z?,      ; ...
U 1264, 0000,013C,01C0,0A08,0000,1084  ;3192   Q_R[1]    ; Get Pointer to Test Patterns
```

```
            ;3193  .PAGE
            ;3194 ;
            ;3195 ;////////////////////////////////////////////////////////////////////////////////
            ;3196 ;
U 1084. 0000,003D,0D80,0800,0084,702A  ;3197 =0 ERROR1[.3]   ; Wrong data returned from the
            ;3198       ; ...ID 1C Register
            ;3199
            ;3200 ;
            ;3201 ;////////////////////////////////////////////////////////////////////////////////
            ;3202 ;
            ;3203 ; ERROR LOGOUT:
            ;3204 ;
            ;3205 ; DATA: EXPECTED DATA FROM ID 1C REGISTER
            ;3206 ;  RECEIVED DATA FROM ID 1C REGISTER
            ;3207 ;  BIT MASK
            ;3208 ;
            ;3209 ;
            ;3210 ;////////////////////////////////////////////////////////////////////////////////
            ;3211 ;
U 1085. 0019,2014,0580,0A88,0000,1265  ;3212  R[1]_Q+K[.1]   ; Point to next Test Pattern in
            ;3213       ; ...Temporary Registers
            ;3214  ALU_R[00]-K[.1],D_ALU.  ; Decrement Loop Counter
U 1265. 0818,0000,0580,0A00,0010,1266  ;3215   CLK.UBCC  ; ...and see if it is 0
U 1266. 0001,013C,0180,0A80,0000,1086  ;3216  Z?,R[00]_D   ; ...save it
U 1086. 0000,003C,0180,0800,0000,1082  ;3217 =0 J/TEST.9.LOOP1   ; Loop Counter NOT 0 yet
            ;3218 TEST.9.EXIT:
U 1087. 0000,003C,0180,0800,0000,1088  ;3219  NOP    ; Finished Test 9
            ;3220
            ;3221
```

```
                ;3222  .PAGE "TEST 192 COMPARATOR REGISTER COUNT FIELD TEST"
                ;3223 ;*************************************************************************
                ;3224 ;**
                ;3225 ;
                ;3226 ;TEST 192 SILO COMPARATOR REGISTER COUNT FIELD TEST
                ;3227 ;
                ;3228 ;TEST DESCRIPTION
                ;3229 ; THIS TEST CHECKS THAT THE SILO COMPARATOR REGISTER
                ;3230 ; COUNT FIELD <19:16> LOADS AND INCREMENTS CORRECTLY.
                ;3231 ; VALUES OF E THRU 2 ARE THE TEST PATTERNS IN THE COUNT
                ;3232 ; FIELD. R3 <3:0> CONTAINS THE COUNT FIELD CURRENTLY
                ;3233 ; BEING TESTED SHOULD AN ERROR OCCUR. THE SILO COMPARATOR
                ;3234 ; REGISTER IS READ 15 TIMES AND PUT INTO THE RC REGISTERS
                ;3235 ; BEGINNING WITH RC E.
                ;3236 ;
                ;3237 ;LOGIC DESCRIPTION
                ;3238 ; THIS TEST CHECKS THE SILO COMPARATOR REGISTER
                ;3239 ; COUNTER, THE SILO LOCK BIT <31>.
                ;3240 ;
                ;3241 ;ERROR DESCRIPTION
                ;3242 ; FIRST READ OF THE COMP REG
                ;3243 ; SECOND READ OF THE COMP REG
                ;3244 ; .
                ;3245 ; .
                ;3246 ; .
                ;3247 ; FIFTEENTH READ OF THE COMP REG
                ;3248 ;
                ;3249 ;--
                ;3250 ;*************************************************************************
                ;3251 ;
                ;3252 =0
U 1088. 0000.003D.0180.0800.0000.117C  ;3253 CNTST: NEWTST
U 1089. 0018.0038.6180.09F8.0000.108A  ;3254  RC[0F]_K[.F]
U 108A. 0F00.003D.0180.0800.0000.11E4  ;3255 =0 CALL,J/DCE,D_0
U 108B. 0001.003C.0180.0A98.0000.108C  ;3256  R[03]_D    ;INITIALIZE COUNT FIELD
U 108C. 0000.003D.0180.0800.0000.119D  ;3257 =0 ERLOOP
U 108D. 0018.0038.61C0.0800.0000.1267  ;3258 NXVAL:  Q_K[.F]
U 1267. 0019.2000.0580.0A88.0000.108E  ;3259  R[01]_Q-K[.1]   ;INITIALIZE RC ADDRESS FOR READ
U 108E. 0000.003D.0180.0800.0000.11CF  ;3260 =0 CALL,J/COMDAT   ;GET DATA FOR COMP REG
U 108F. 0000.003C.7180.3C00.0000.1268  ;3261  ID[COMP]_D   ;WRITE COMP REG
U 1268. 0000.003C.71F0.2C00.0000.1269  ;3262  Q_ID[COMP]   ;READ COMP REG AND PUT
U 1269. 0001.203C.71F0.2DF0.0000.126A  ;3263  RC[0E]_Q,Q_ID[COMP]  ;INTO RC REGISTERS
U 126A. 0001.203C.71F0.2DE8.0000.126B  ;3264  RC[0D]_Q,Q_ID[COMP]
U 126B. 0001.203C.71F0.2DE0.0000.126C  ;3265  RC[0C]_Q,Q_ID[COMP]
U 126C. 0001.203C.71F0.2DD8.0000.126D  ;3266  RC[0B]_Q,Q_ID[COMP]
U 126D. 0001.203C.71F0.2DD0.0000.126E  ;3267  RC[0A]_Q,Q_ID[COMP]
U 126E. 0001.203C.71F0.2DC8.0000.126F  ;3268  RC[09]_Q,Q_ID[COMP]
U 126F. 0001.203C.71F0.2DC0.0000.1270  ;3269  RC[08]_Q,Q_ID[COMP]
U 1270. 0001.203C.71F0.2DB8.0000.1271  ;3270  RC[07]_Q,Q_ID[COMP]
U 1271. 0001.203C.71F0.2DB0.0000.1272  ;3271  RC[06]_Q,Q_ID[COMP]
U 1272. 0001.203C.71F0.2DA8.0000.1273  ;3272  RC[05]_Q,Q_ID[COMP]
U 1273. 0001.203C.71F0.2DA0.0000.1274  ;3273  RC[04]_Q,Q_ID[COMP]
U 1274. 0001.203C.71F0.2D98.0000.1275  ;3274  RC[03]_Q,Q_ID[COMP]
U 1275. 0001.203C.71F0.2D90.0000.1276  ;3275  RC[02]_Q,Q_ID[COMP]
U 1276. 0001.203C.71F0.2D88.0000.1277  ;3276  RC[01]_Q,Q_ID[COMP]
```

```
U 1277. 0001.203C.0180.0980.0000.1278  ;3277  RC[0]_Q
U 1278. 0000.003C.01C0.0A18.0000.1279  ;3278  Q_R[03]
U 1279. 0001.203C.0180.0A80.0000.127A  ;3279 NXRC:  R[0]_Q     ;EXPECTED COUNT FIELD
U 127A. 0818.0038.6580.0A00.0000.127B  ;3280  D_K[.10],LAB_R[0]
U 127B. 001C.2000.0180.0800.0010.127C  ;3281  ALU_LA-D,CLK.UBCC
U 127C. 0000.013C.0180.0800.0000.1090  ;3282  Z?
U 1090. 0000.003C.0180.0800.0000.1286  ;3283 =0 J/RDRC     ;LOCK BIT SHOULDN'T BE SET
U 1091. 0000.003C.0180.0800.0000.1092  ;3284  NOP
U 1092. 0000.003D.0180.0800.0000.11C6  ;3285 =0 CALL,J/LKCYC    ;GET EXPECTED DATA FOR
U 1093. 0001.003C.0180.0A90.0000.127D  ;3286  R[02]_D     ;LOCK BIT SET CYCLES
U 127D. 0000.003C.0180.0A08.0082.127E  ;3287 ALLOCK: SC_R[01]
U 127E. 0000.003C.0180.0830.0000.127F  ;3288  LC_RC(SC)
U 127F. 0800.003C.0180.0A10.0000.1280  ;3289  D_R[02]
U 1280. 0011.0000.0180.0800.0010.1281  ;3290  ALU_D-LC,CLK.UBCC
U 1281. 0000.013C.0180.0800.0000.1094  ;3291  Z?
U 1094. 0000.003D.0180.0800.0000.119E  ;3292 =0 ERROR1     ;COMP REG DATA INCORRECT
U 1095. 0818.0038.1D80.0800.0010.1282  ;3293  ALU_SC,D_ALU,CLK.UBCC
U 1282. 0000.013C.01C0.0A18.0000.1096  ;3294  Z?,Q_R[03]    ;LAST RC?
U 1096. 0019.0000.0580.0A88.0000.127D  ;3295 =0 R[01]_D-K[.1],J/ALLOCK  ;DECREMENT RC ADD,READ NEXT
U 1097. 0019.2000.0580.0A98.0000.1283  ;3296  R[03]_Q-K[.1]    ;DECREMENT COUNT FIELD
U 1283. 0818.0038.0580.0A18.0000.1284  ;3297  D_K[.1],LAB_R[03]
U 1284. 001C.2000.0180.0800.0010.1285  ;3298  ALU_LA-D,CLK.UBCC
U 1285. 0000.013C.0180.0800.0000.1098  ;3299  Z?
U 1098. 0000.003C.0180.0800.0000.108D  ;3300 =0 J/NXVAL     ;CHECK NEXT COUNT FIELD
U 1099. 0000.003C.0180.0800.0000.1292  ;3301  J/FINIS     ;ALL COUNT FIELDS TESTED
U 1286. 0800.003C.0180.0A00.0000.1287  ;3302 RDRC:  D_R[0]
U 1287. 0000.003C.65F8.0800.0084.7288  ;3303  SC_K[.10],Q_0
U 1288. 0D00.003C.0180.0800.0000.1289  ;3304  D_DAL.SC    ;SET EXPECTED COUNT FIELD
U 1289. 0018.0038.51C0.0800.0000.128A  ;3305  Q_K[.1E]
U 128A. 0019.2000.0580.0800.0082.128B  ;3306  SC_Q-K[.1]
U 128B. 0801.001C.0180.0800.0000.128C  ;3307  D_D.ORNOT.MASK    ;SET <29>, UNCOND LOCK
U 128C. 0000.003C.0180.0A08.0082.128D  ;3308  SC_R[01]
U 128D. 0000.003C.0180.0830.0000.128E  ;3309  LC_RC(SC)    ;GET RECEIVED DATA
U 128E. 0011.0000.0180.0800.0010.128F  ;3310  ALU_D-LC,CLK.UBCC
U 128F. 0000.013C.01C0.0A08.0000.109A  ;3311  Z?,Q_R[01]
U 109A. 0000.003D.0180.0800.0000.119E  ;3312 =0 ERROR1     ;COMP REG DATA INCORRECT
U 109B. 0019.2000.0580.0A88.0000.1290  ;3313  R[01]_Q-K[.1]    ;DECREMENT RC ADDRESS
U 1290. 0800.003C.0180.0A00.0000.1291  ;3314  D_R[0]
U 1291. 0019.0014.05C0.0800.0000.1279  ;3315  Q_D+K[.1],J/NXRC  ;INCREMENT COUNT FIELD
U 1292. 0000.003C.0180.0800.0000.109C  ;3316 FINIS: NOP
            ;3317
            ;3318
```

```
;3319  .PAGE "TEST 193 SILO LOCK ON ID COMPARE TEST"
;3320 ;*************************************************************************
;3321 ;**
;3322 ;
;3323 ;TEST 195  SILO LOCK ON ID COMPARE TEST
;3324 ;
;3325 ;TEST DESCRIPTION
;3326 ; THIS TEST CHECKS THAT THE SILO LOCKS CORRECTLY WHEN
;3327 ; THE SILO COMPARATOR REGISTER CONDITIONAL LOCK BITS<28:27>
;3328 ; ARE EQUAL TO 01 (LOCK ON ID ONLY).
;3329 ;
;3330 ; SUBTEST 1 PUTS THE CPU'S ID INTO THE MAINTENANCE
;3331 ;   REGISTER ID <27:23> AND CHECKS THAT THE
;3332 ;   SILO LOCKS WHEN A CPU WRITE TO MEMORY IS
;3333 ;   DONE (MAINTENANCE ID=SBI ID) BY CHECKING
;3334 ;   THE SILO COMPARATOR REGISTER.
;3335 ;
;3336 ;
;3337 ; SUBTEST 2 PUTS FIVE DIFFERENT DATA PATTERNS IN
;3338 ;   THE MAINTENANCE REGISTER ID <27:23> AND
;3339 ;   THEN DOES A CPU WRITE TO MEMORY. IT THEN
;3340 ;   CHECKS THAT THE SILO DOES NOT LOCK AND THE
;3341 ;   COUNT FIELD REMAINS UNCHANGED BY
;3342 ;   EXAMINING ID 1C. IF A DATA PATTERN IS
;3343 ;   EQUAL TO THE CPU'S ID, IT IS NOT TESTED HERE.
;3344 ;
;3345 ;   DATA PATTERNS:  00001
;3346 ;     00010
;3347 ;     00100
;3348 ;     01000
;3349 ;     10000
;3350 ;
;3351 ; SUBTEST 3 USES THE SAME DATA PATTERNS AS SUBTEST 2.
;3352 ;   FORCE MULTIPLE TRANSMITTERS IS SET IN THE
;3353 ;   MAINTENANCE REGISTER <28>, FORCING THE
;3354 ;   MAINTENANCE ID TO BE TRANSMITTED WITH THE
;3355 ;   WRITE DATA,THUS CAUSING AN ID MATCH. THE
;3356 ;   COMPARE SIGNAL ASSERTING IS DETECTED ONLY BY
;3357 ;   THE COUNT FIELD INCREMENTING SINCE THE SILO
;3358 ;   COMPARATOR LOCK BIT <31> IS DISABLED BY AN SBI
;3359 ;   FAULT. THE CPU'S ID IS NOT TESTED HERE.
;3360 ;
;3361 ;LOGIC DESCRIPTION
;3362 ;
;3363 ;  N/A
;3364 ;
;3365 ;ERROR DESCRIPTION
;3366 ; EXPECTED SILO COMPARATOR REGISTER DATA
;3367 ; RECEIVED SILO COMPARATOR REGISTER DATA
;3368 ; ID UNDER TEST IN BITS <4:0>
;3369 ;
;3370 ;-
;3371 ;*************************************************************************
;3372 ;
;3373 =0
```

```
U 109C. 0000.003D.0180.0800.0000.117C  ;3374 LKTST: NEWTST
U 109D. 0018.0038.0D80.09F8.0000.109E  ;3375  RC[0F]_K[.3]
U 109E. 0000.003D.0180.0800.0000.1255  ;3376 =0 CALL[CONFIG.MS780E]  ; Make sure MS780-E/H is properly
          ;3377       ; ...configured
U 109F. 0000.003C.0180.0800.0000.10A0  ;3378  NOP
          ;3379
          ;3380 ; SUBTEST 1
          ;3381 ;
U 10A0. 0000.003D.0180.0800.0000.11A2  ;3382 =0 SUBTEST
U 10A1. 0000.003C.0180.0800.0000.10A2  ;3383  NOP
U 10A2. 0000.003D.0180.0800.0000.1036  ;3384 =0 CALL,J/CPUCA    ;GET CPU'S ID
U 10A3. 0B00.003C.0180.0800.0000.1293  ;3385  D_D.SWAP
U 1293. 0600.003C.0180.0800.0000.1294  ;3386  D_D.RIGHT
U 1294. 0819.0034.8D80.0800.0000.1295  ;3387  D_D.AND.K[.1F]    ;CPU'S ID <4:0>
U 1295. 0001.003C.0180.09E0.0000.1296  ;3388  RC[0C]_D    ;SAVE
U 1296. 0001.003C.0180.0980.0000.10A4  ;3389  RC[0]_D
U 10A4. 0000.003D.0180.0800.0000.119D  ;3390 =0 ERLOOP
U 10A5. 0000.003C.8980.0800.0084.7297  ;3391  SC_K[.D]
U 1297. 0003.001C.0180.0AF8.0000.10A6  ;3392  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 10A6. 0000.003D.0180.0800.0000.11F0  ;3393 =0 CALL,J/STCOMP
U 10A7. 0000.003C.7180.3C00.0000.10A8  ;3394  ID[COMP]_D    ;SET TO LOCK ON ID ONLY
U 10A8. 0000.003D.0180.0800.0000.11EB  ;3395 =0 CALL,J/EXCOMP    ;GET EXPECTED DATA
U 10A9. 0001.003C.0180.09F0.0000.10AA  ;3396  RC[0E]_D
U 10AA. 0000.003D.0180.0800.0000.11F2  ;3397 =0 CALL,J/SETMRG
U 10AB. 0000.003C.7580.3C00.0000.1298  ;3398  ID[MAINT]_D    ;PUT CPU'S ID IN <27:23>
U 1298. 0018.0038.1980.0800.0200.1299  ;3399  VA_K[ZERO]
U 1299. 0000.003C.0180.A800.0000.129A  ;3400  MCT/WRITE.P    ;PUT ID ON SBI
U 129A. 0000.003C.0180.0800.0000.129B  ;3401  NOP
U 129B. 0000.003C.0180.0800.0000.129C  ;3402  NOP
U 129C. 0000.003C.0180.0800.0000.129D  ;3403  NOP
U 129D. 0000.003C.71F0.2C00.0000.129E  ;3404  Q_ID[COMP]    ;READ COMP REG
U 129E. 0001.203C.0180.09E8.0000.129F  ;3405  RC[0D]_Q
U 129F. 0810.0038.0180.0970.0000.12A0  ;3406  D_RC[0E]
U 12A0. 001D.0000.0180.0800.0010.12A1  ;3407  ALU_D-Q,CLK.UBCC
U 12A1. 0000.013C.0180.0800.0000.10AC  ;3408  Z?
U 10AC. 0000.003D.0180.0800.0000.11A0  ;3409 =0 ERROR2     ;COMP REG DATA INCORRECT
U 10AD. 0000.003C.0180.0800.0000.10AE  ;3410  NOP     ;LOCKED OK ON CPU'S ID!
          ;3411 ;
          ;3412 ;
          ;3413 ; SUBTEST 2
          ;3414 ;
U 10AE. 0000.003D.0180.0800.0000.11A2  ;3415 =0 SUBTEST
U 10AF. 0018.0038.0D80.09F8.0000.10B0  ;3416  RC[0F]_K[.3]
U 10B0. 0000.003D.0180.0800.0000.1036  ;3417 =0 CALL,J/CPUCA    ;GET CPU'S ID
U 10B1. 0B00.003C.0180.0800.0000.12A2  ;3418  D_D.SWAP
U 12A2. 0600.003C.0180.0800.0000.12A3  ;3419  D_D.RIGHT
U 12A3. 0819.0034.8D80.0800.0000.12A4  ;3420  D_D.AND.K[.1F]
U 12A4. 0001.003C.0180.0990.0000.12A5  ;3421  RC[02]_D    ;SAVE CPU ID IN <4:0>
U 12A5. 0818.0038.0580.0800.0000.12A6  ;3422  D_K[.1]
U 12A6. 0001.003C.0180.09E0.0000.12A7  ;3423    RC[0C]_D
U 12A7. 0001.003C.0180.0980.0000.12A8  ;3424 CHKID: RC[0]_D     ;SAVE
U 12A8. 0010.0038.01C0.0910.0000.12A9  ;3425  Q_RC[02]
U 12A9. 001D.0000.0180.0800.0010.12AB  ;3426  ALU_D-Q,CLK.UBCC
U 12AB. 0000.013C.0180.0800.0000.10B2  ;3427  Z?     ;CPU'S ID?
U 10B2. 0000.003C.0180.0800.0000.10B5  ;3428 =0 J/GOMREG
```

```
U 10B3, 0000.003C.0180.0800.0000.10BB  ;3429  J/NXTID    ;CPU'S ID, DO NOT TEST
U 10B4, 0000.003D.0180.0800.0000.119D  ;3430 =0  ERLOOP
U 10B5, 0000.003C.8980.0800.0084.72AC  ;3431 COMREG: SC_K[.D]
U 12AC, 0003.001C.0180.0AF8.0000.10B6  ;3432  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 10B6, 0000.003D.0180.0800.0000.11F2  ;3433 =0 CALL,J/SETMRG
U 10B7, 0000.003C.7580.3C00.0000.10B8  ;3434  ID[MAINT]_D   ;TEST ID IN BITS <27:23>
U 10B8, 0000.003D.0180.0800.0000.11F0  ;3435 =0 CALL,J/STCOMP
U 10B9, 0000.003C.7180.3C00.0000.12AD  ;3436  ID[COMP]_D   ;SET TO LOCK ON ID ONLY
U 12AD, 0001.003C.0180.09F0.0000.12AE  ;3437  RC[0E]_D   ;SAVE
U 12AE, 0018.0038.1980.0800.0200.12AF  ;3438  VA_K[ZERO]
U 12AF, 0000.003C.0180.A800.0000.12B0  ;3439  MCT/WRITE.P
U 12B0, 0000.003C.0180.0800.0000.12B1  ;3440  NOP
U 12B1, 0000.003C.0180.0800.0000.12B2  ;3441  NOP
U 12B2, 0000.003C.0180.0800.0000.12B3  ;3442  NOP
U 12B3, 0000.003C.71F0.2C00.0000.12B4  ;3443  Q_ID[COMP]   ;READ COMP REG
U 12B4, 0001.203C.0180.09E8.0000.12B5  ;3444  RC[0D]_Q
U 12B5, 0810.0038.0180.0970.0000.12B6  ;3445  D_RC[0E]
U 12B6, 001D.0000.0180.0800.0010.12B7  ;3446  ALU_D-Q,CLK.UBCC
U 12B7, 0000.013C.0180.0800.0000.10BA  ;3447  Z?
U 10BA, 0000.003D.0180.0800.0000.11A0  ;3448 =0 ERROR2    ;COMP REG DATA INCORRECT
U 10BB, 0810.0038.0180.0900.0000.12B8  ;3449 NXTID: D_RC[0]
U 12B8, 0500.003C.0180.0800.0000.12B9  ;3450  D_D.LEFT.SI/ZERO  ;GENERATE NEXT TEST ID
U 12B9, 0019.0034.8D80.09E0.0000.12BA  ;3451  RC[0C]_D.AND.K[.1F]
U 12BA, 0010.0038.0180.0960.0010.12BB  ;3452  ALU_RC[0C],CLK.UBCC
U 12BB, 0000.013C.0180.0800.0000.10BC  ;3453  Z?    ;LAST TEST PATTERN?
U 10BC, 0000.003C.0180.0800.0000.12A7  ;3454 =0 J/CHKID    ;TEST NEXT ID
U 10BD, 0000.003C.0180.0800.0000.10BE  ;3455  NOP    ;DONE
          ;3456 ;
          ;3457 ; SUBTEST 3
U 10BE, 0000.003D.0180.0B00.0000.11A2  ;3458 =0 SUBTEST
U 10BF, 0018.0038.0D80.09F8.0000.10C0  ;3459  RC[0F]_K[.3]
U 10C0, 0000.003D.0180.0800.0000.1036  ;3460 =0 CALL,J/CPUCA
U 10C1, 0B00.003C.0180.0800.0000.12BC  ;3461  D_D.SWAP
U 12BC, 0600.003C.0180.0800.0000.12BD  ;3462  D_D.RIGHT
U 12BD, 0819.0034.8D80.0800.0000.12BE  ;3463  D_D.AND.K[.1F]
U 12BE, 0001.003C.0180.0990.0000.12BF  ;3464  RC[02]_D    ;SAVE CPU ID IN <4:0>
U 12BF, 0818.0038.0580.0800.0000.12C0  ;3465  D_K[.1]
U 12C0, 0001.003C.0180.09E0.0000.12C1  ;3466  RC[0C]_D
U 12C1, 0001.003C.0180.0980.0000.12C2  ;3467 CHNXT: RC[0]_D
U 12C2, 0010.0038.01C0.0910.0000.12C3  ;3468  Q_RC[02]
U 12C3, 001D.0000.0180.0800.0010.12C4  ;3469  ALU_D-Q,CLK.UBCC
U 12C4, 0000.013C.0180.0800.0000.10C2  ;3470  Z?
U 10C2, 0000.003C.0180.0800.0000.10C5  ;3471 =0 J/MREG
U 10C3, 0000.003C.0180.0800.0000.10CF  ;3472  J/NEWID    ;TEST PAT=CPU ID DON'T TEST
U 10C4, 0000.003D.0180.0800.0000.119D  ;3473 =0  ERLOOP
U 10C5, 0000.003C.8980.0800.0084.72C5  ;3474 MREG: SC_K[.D]
U 12C5, 0003.001C.0180.0AF8.0000.12C6  ;3475  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 12C6, 0000.003C.2180.0800.0084.72C7  ;3476  SC_K[.14]
U 12C7, 0000.003C.0580.0800.0084.B2C8  ;3477  SC_SC-K[.1]
U 12C8, 0803.001C.0180.0800.0000.12C9  ;3478  D_NOT.MASK
U 12C9, 0001.003C.0180.0A98.0000.12CA  ;3479  R[3]_D
U 12CA, 0F00.003C.6D80.3C00.0000.12CB  ;3480  ID[FAULT]_D,D_0    ;CLEAR THE FAULT REG
U 12CB, 0000.003C.7580.3C00.0000.10C6  ;3481  ID[MAINT]_D    ;CLEAR THE MAINT REG
U 10C6, 0000.003D.0180.0800.0000.11AE  ;3482 =0 CALL,J/HLD.UNJ    ;UNJAM THE SBI
U 10C7, 0000.003C.0180.0800.0000.10C8  ;3483  NOP
```

```
U 10C8. 0000.003D.0180.0800.0000.11F2  ;3484 =0 CALL,J/SETMRG
U 10C9. 0018.0038.51C0.0800.0000.12CC  ;3485  Q_K[.1E]
U 12CC. 0019.2000.0980.0800.0082.12CD  ;3486  SC_Q-K[.2]
U 12CD. 0801.001C.0180.0800.0000.12CE  ;3487  D_D.ORNOT.MASK    ;SET FORCE MULT TRANS <28>
U 12CE. 0000.003C.7580.3C00.0000.10CA  ;3488  ID[MAINT]_D
U 10CA. 0000.003D.0180.0800.0000.11F0  ;3489 =0 CALL,J/STCOMP
U 10CB. 0000.003C.7180.3C00.0000.12CF  ;3490  ID[COMP]_D    ;SET UP COMP REG
U 12CF. 0000.003C.6580.0800.0084.72D0  ;3491  SC_K[.10]
U 12D0. 0801.001C.0180.0800.0000.12D1  ;3492  D_D.ORNOT.MASK    ;SET EXPECTED COUNT=F
U 12D1. 0001.003C.0180.09F0.0000.12D2  ;3493  RC[0E]_D
U 12D2. 0018.0038.1980.0800.0200.12D3  ;3494  VA_K[ZERO]
U 12D3. 0000.003C.0180.A800.0000.12D4  ;3495  MCT/WRITE.P
U 12D4. 0000.003C.0180.0800.0000.12D5  ;3496  NOP     ;WAIT FOR WRITE TO COMPLETE
U 12D5. 0000.003C.0180.0800.0000.12D6  ;3497  NOP
U 12D6. 0000.003C.0180.0800.0000.12D7  ;3498  NOP
U 12D7. 0000.003C.0180.0800.0000.12D8  ;3499  NOP
U 12D8. 0000.003C.0180.0800.0000.12D9  ;3500  NOP
U 12D9. 0000.003C.71F0.2C00.0000.12DA  ;3501  Q_ID[COMP]    ;READ COMP REG
U 12DA. 0001.203C.0180.09D8.0000.12DB  ;3502  RC[0B]_Q
U 12DB. 0818.0038.8180.0800.0000.10CC  ;3503  D_K[.3FF]
U 10CC. 0000.003D.0180.0800.0000.11E5  ;3504 =0 CALL,J/DCF
U 10CD. 0000.003C.65F8.0800.0084.72DC  ;3505  SC_K[.10],Q_0
U 12DC. 0D00.003C.0180.0800.0000.12DD  ;3506  D_DAL.SC
U 12DD. 0000.003C.5180.0800.0084.72DE  ;3507  SC_K[.1E]
U 12DE. 0801.001C.0180.0800.0000.12DF  ;3508  D_D.ORNOT.MASK
U 12DF. 0010.0038.01C0.0958.0000.12E0  ;3509  Q_RC[0B]
U 12E0. 081D.0034.0180.0800.0000.12E1  ;3510  D_D.AND.Q    ;IGNORE LOCK BIT <31>
U 12E1. 0001.003C.0180.09E8.0000.12E2  ;3511  RC[0D]_D
U 12E2. 0010.0038.01C0.0970.0000.12E3  ;3512  Q_RC[0E]
U 12E3. 001D.0000.0180.0800.0010.12E4  ;3513  ALU_D-Q,CLK.UBCC
U 12E4. 0000.013C.0180.0800.0000.10CE  ;3514  Z?
U 10CE. 0000.003D.0180.0800.0000.11A0  ;3515 =0 ERROR2     ;COMP REG DATA INCORRCT
U 10CF. 0810.0038.0180.0900.0000.12E5  ;3516 NEWID: D_RC[0]
U 12E5. 0500.003C.0180.0800.0000.12E6  ;3517  D_D.LEFT,SI/ZERO   ;NEW TEST PATTERN
U 12E6. 0019.0034.8D80.09E0.0000.12E7  ;3518  RC[0C]_D.AND.K[.1F]
U 12E7. 0010.0038.0180.0960.0010.12E8  ;3519  ALU_RC[0C],CLK.UBCC
U 12E8. 0000.013C.0180.0800.0000.10D0  ;3520  Z?     ;DONE YET?
U 10D0. 0000.003C.0180.0800.0000.12C1  ;3521 =0 J/CHNXT     ;GO TEST NEXT PATTERN
U 10D1. 0800.003C.0180.0A18.0000.12E9  ;3522  D_R[3]
U 12E9. 0F00.003C.6D80.3C00.0000.12EA  ;3523  ID[FAULT]_D,D_0    ;CLEAR THE FAULT
U 12EA. 0000.003C.7580.3C00.0000.10D2  ;3524  ID[MAINT]_D    ;CLEAR MULT XMITTERS
            ;3525
```

```
;3526  .PAGE "TEST 194 SILO LOCK ON ID & TAG COMPARE TEST"
;3527 ;*****************************************************************
;3528 ;**
;3529 ;
;3530 ;TEST 194  SILO LOCK ON ID & TAG COMPARE TEST
;3531 ;
;3532 ;TEST DESCRIPTION
;3533 ; THIS TEST CHECKS THAT THE SILO LOCKS CORRECTLY WHEN THE
;3534 ; SILO COMPARATOR REGISTER CONDITIONAL LOCK BITS <28:27> ARE
;3535 ; EQUAL TO 10 (LOCK ON ID AND TAG COMPARISON). THE SILO LOCK
;3536 ; ON COMPARISON FUNCTION IS CHECKED BY EXAMINING THE SILO
;3537 ; COMPARATOR REGISTER.
;3538 ;
;3539 ; SUBTEST 1 USES THE CPU'S ID AND A READ DATA TAG.
;3540 ;   A CPU READ OF MEMORY IS EXECUTED. THE
;3541 ;   SILO SHOULD LOCK.
;3542 ;
;3543 ; SUBTEST 2 USES THE CPU'S ID AND A COMMAND ADDRESS
;3544 ;   TAG. A CPU WRITE TO MEMORY IS EXECUTED.
;3545 ;   THE SILO SHOULD LOCK.
;3546 ;
;3547 ; SUBTEST 3 USES THE CPU'S ID AND A WRITE DATA
;3548 ;   TAG. A CPU WRITE TO MEMORY IS EXECUTED.
;3549 ;   THE SILO SHOULD LOCK.
;3550 ;
;3551 ; SUBTEST 4 USES THE CPU'S ID AND AN ISR TAG. AN
;3552 ;   INTERRUPT SUMMARY READ IS INITIATED.
;3553 ;   THE SILO SHOULD LOCK.
;3554 ;
;3555 ; SUBTEST 5 USES AN ID WHICH IS NOT THE CPU'S ID
;3556 ;   AND A COMMAND ADDRESS TAG. A CPU WRITE
;3557 ;   TO MEMORY IS EXECUTED. THE SILO SHOULD
;3558 ;   NOT LOCK.
;3559 ;
;3560 ; SUBTEST 6 USES THE CPU'S ID AND A READ DATA TAG.
;3561 ;   A CPU WRITE TO MEMORY IS EXECUTED.
;3562 ;   THE SILO SHOULD NOT LOCK.
;3563 ;
;3564 ;LOGIC DESCRIPTION
;3565 ;
;3566 ;  N/A
;3567 ;
;3568 ;ERROR DESCRIPTION
;3569 ; DATA: EXPECTED SILO COMPARATOR DATA
;3570 ;  RECEIVED SILO COMPARATOR DATA
;3571 ;  TEST TAG <3:0>
;3572 ;  TEST ID <4:0>
;3573 ;
;3574 ;--
;3575 ;*****************************************************************
;3576 ;
;3577 =0
U 10D2, 0000,003D,0180,0800,0000,117C  ;3578 LKIDT: NEWTST
U 10D3, 0018,0038,1180,09F8,0000,10D4  ;3579  RC[0F]_K[.4]
U 10D4, 0000,003D,0180,0800,0000,1255  ;3580 =0 CALL[CONFIG.MS780E]  ; Make sure that MS780-E/H is properly
```

```
            ;3581       ; ...configured
U 10D5. 0818.0038.7580.0800.0000.12EB   ;3582  D_K[.20]
U 12EB. 0B00.003C.0180.0800.0000.12EC   ;3583  D_D.SWAP
U 12EC. 0818.0038.75E0.0800.0000.10D6   ;3584  Q_D,D_K[.20]
U 10D6. 0000.003D.0180.0800.0000.11D6   ;3585 =0 CALL[DC0]
U 10D7. 0000.003C.0180.0800.0000.10D8   ;3586  NOP
U 10D8. 0000.003D.0180.0800.0000.11D6   ;3587 =0 CALL[DC0]
U 10D9. 0001.003C.0180.0980.0000.12ED   ;3588  RC[0]_D     ; D=2000
U 12ED. 001D.0030.0180.0A80.0000.12EE   ;3589  R[0]_D.OR.Q    ; R0=20002000
U 12EE. 0018.0038.6180.0A90.0000.12EF   ;3590  R[2]_K[.F]    ; LOOP COUNT
U 12EF. 0000.003C.8980.0800.0084.72F0   ;3591 NEWTR: SC_K[.D]
U 12F0. 0003.001C.0180.0AF8.0000.12F1   ;3592  R[0F]_NOT.MASK    ; SET TO CATCH TIMEOUT
U 12F1. 0000.003C.0180.0A00.0200.12F2   ;3593  VA_R[0]
U 12F2. 0000.003C.0180.C800.0000.12F3   ;3594  D[LONG]_CACHE.P    ; REFERENCE THE LOCATION
U 12F3. 0000.003C.0180.0A78.0010.12F4   ;3595  ALU_R[0F],CLK.UBCC   ; CHECK FOR TIMEOUT
U 12F4. 0000.013C.0180.0900.0000.10DA   ;3596  Z?,LC_RC[0]
U 10DA. 0000.003C.0180.0800.0000.12FD   ;3597 =0 J/DOTEST
U 10DB. 0818.0038.4580.0800.0000.12F5   ;3598  D_K[.8000]
U 12F5. 0200.003C.0180.0800.0000.12F6   ;3599  D_D.RIGHT2
U 12F6. 0600.003C.0180.0800.0000.12F7   ;3600  D_D.RIGHT
U 12F7. 0000.003C.6580.3C00.0000.12F8   ;3601  ID[SBI.ERR]_D    ; CLEAR THE TIMEOUT
U 12F8. 0800.003C.0180.0A00.0000.12F9   ;3602  D_R[0]
U 12F9. 0811.0014.0180.0A80.0000.12FA   ;3603  D_D+LC,R[0]_ALU    ; R0=ADDR OF CONFIG REG
U 12FA. 0000.003C.0180.0A10.0000.12FB   ;3604  LAB_R[2]    ; GET THE LOOP COUNT
U 12FB. 0018.0000.0580.0A90.0010.12FC   ;3605  R[2]_LA-K[.1],CLK.UBCC   ; LAST TR?
U 12FC. 0000.013C.0180.0800.0000.10DC   ;3606  Z?
U 10DC. 0000.003C.0180.0800.0000.12EF   ;3607 =0 J/NEWTR
U 10DD. 0000.003C.0180.0800.0000.112B   ;3608  J/NODEVS    ; NO SBI DEVICES AVAILABLE
U 12FD. 0000.003C.0180.0800.0000.10DE   ;3609 DOTEST: NOP
U 10DE. 0000.003D.0180.0800.0000.1036   ;3610 =0 CALL,J/CPUCA
U 10DF. 0B00.003C.0180.0800.0000.12FE   ;3611  D_D.SWAP
U 12FE. 0600.003C.0180.0800.0000.12FF   ;3612  D_D.RIGHT
U 12FF. 0819.0034.8D80.0800.0000.1300   ;3613  D_D.AND.K[.1F]
U 1300. 0001.003C.0180.0980.0000.10E0   ;3614  RC[0]_D
            ;3615 ;
            ;3616 ; SUBTEST 1
            ;3617 ;
U 10E0. 0000.003D.0180.0800.0000.11A2   ;3618 =0 SUBTEST
U 10E1. 0000.003C.0180.0800.0000.10E2   ;3619  NOP
U 10E2. 0000.003D.0180.0800.0000.119D   ;3620 =0 ERLOOP
U 10E3. 0810.0038.0180.0900.0000.1301   ;3621  D_RC[0]     ;GET CPU'S ID
U 1301. 0001.003C.0180.09D8.0000.10E4   ;3622  RC[0B]_D    ;TEST ID <4:0>
U 10E4. 0000.003D.0180.0800.0000.11F2   ;3623 =0 CALL,J/SETMRG
U 10E5. 0000.003C.7580.3C00.0000.1302   ;3624  ID[MAINT]_D
U 1302. 0018.0038.1980.09E0.0000.10E6   ;3625  RC[0C]_K[ZERO]    ;READ DATA TAG
U 10E6. 0000.003D.0180.0800.0000.11F9   ;3626 =0 CALL,J/TGCOMP
U 10E7. 0000.003C.7180.3C00.0000.1303   ;3627  ID[COMP]_D    ;SET UP COMP REG
U 1303. 0000.003C.8D80.0800.0084.7304   ;3628  SC_K[.1F]
U 1304. 0801.001C.0180.0800.0000.1305   ;3629  D_D.ORNOT.MASK    ;SET <31> COMP LOCK BIT
U 1305. 0000.003C.6580.0800.0084.7306   ;3630  SC_K[.10]
U 1306. 0801.001C.0180.0800.0000.1307   ;3631  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 1307. 0001.003C.0180.09F0.0000.1308   ;3632  RC[0E]_D
U 1308. 0000.003C.8980.0800.0084.7309   ;3633  SC_K[.D]
U 1309. 0003.001C.0180.0AF8.0000.130A   ;3634  R[0F]_ALU,ALU_NOT.MASK   ;SET T.O. TRAP FLAG
U 130A. 0000.003C.0180.0A00.0200.130B   ;3635  VA_R[0]
```

```
U 130B. 0000.003C.0180.C800.0000.130C  ;3636  MCT/READ.P    ;START SBI CYCLE WHICH WILL
U 130C. 0000.003C.0180.0800.0000.130D  ;3637  NOP     ;INITIATE A COMPARE LOCK ON
U 130D. 0000.003C.0180.0800.0000.130E  ;3638  NOP     ;ID AND TAG
U 130E. 0000.003C.0180.0800.0000.130F  ;3639  NOP
U 130F. 0000.003C.0180.0800.0000.1310  ;3640  NOP
U 1310. 0000.003C.71F0.2C00.0000.1311  ;3641  Q_ID[COMP]    ;READ COMP REG
U 1311. 0001.203C.0180.09E8.0000.1312  ;3642  RC[0D]_Q    ;SAVE RECEIVED DATA
U 1312. 0810.0038.0180.0970.0000.1313  ;3643  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 1313. 001D.0000.0180.0800.0010.1314  ;3644  ALU_D-Q,CLK.UBCC
U 1314. 0000.013C.0180.0800.0000.10E8  ;3645  Z?
U 10E8. 0000.003D.0180.0800.0000.11A0  ;3646 =0 ERROR2     ;COMP REG DATA INCORRECT
U 10E9. 0000.003C.0180.0800.0000.10EA  ;3647  NOP     ;DONE
           ;3648 ;
           ;3649 ; SUBTEST 2
           ;3650 ;
U 10EA. 0000.003D.0180.0800.0000.11A2  ;3651 =0 SUBTEST
U 10EB. 0000.003C.0180.0800.0000.10EC  ;3652  NOP
U 10EC. 0000.003D.0180.0800.0000.119D  ;3653 =0 ERLOOP
U 10ED. 0810.0038.0180.0900.0000.1315  ;3654  D_RC[0]     ;GET CPU'S ID
U 1315. 0001.003C.0180.09D8.0000.10EE  ;3655  RC[0B]_D    ;TEST ID <4:0>
U 10EE. 0000.003D.0180.0800.0000.11F2  ;3656 =0 CALL,J/SETMRG
U 10EF. 0000.003C.7580.3C00.0000.1316  ;3657  ID[MAINT]_D
U 1316. 0018.0038.0D80.09E0.0000.10F0  ;3658  RC[0C]_K[.3]    ;COMMAND ADDRESS TAG
U 10F0. 0000.003D.0180.0800.0000.11F9  ;3659 =0 CALL,J/TGCOMP
U 10F1. 0000.003C.7180.3C00.0000.1317  ;3660  ID[COMP]_D    ;SET UP COMP REG
U 1317. 0000.003C.8D80.0800.0084.7318  ;3661  SC_K[.1F]
U 1318. 0801.001C.0180.0800.0000.1319  ;3662  D_D.ORNOT.MASK    ;SET <31> COMP LOCK BIT
U 1319. 0000.003C.6580.0800.0084.731A  ;3663  SC_K[.10]
U 131A. 0801.001C.0180.0800.0000.131B  ;3664  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 131B. 0001.003C.0180.09F0.0000.131C  ;3665  RC[0E]_D    ;SAVE EXPECTED DATA
U 131C. 0000.003C.8980.0800.0084.731D  ;3666  SC_K[.D]
U 131D. 0003.001C.0180.0AF8.0000.131E  ;3667  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 131E. 0F00.003C.0180.0A00.0200.131F  ;3668  VA_R[0],D_0
U 131F. 0000.003C.0180.A800.0000.1320  ;3669  MCT/WRITE.P    ;START SBI CYCLE FOR COMPARE
U 1320. 0000.003C.0180.0800.0000.1321  ;3670  NOP
U 1321. 0000.003C.0180.0800.0000.1322  ;3671  NOP
U 1322. 0000.003C.0180.0800.0000.1323  ;3672  NOP
U 1323. 0000.003C.0180.0800.0000.1324  ;3673  NOP
U 1324. 0000.003C.71F0.2C00.0000.1325  ;3674  Q_ID[COMP]    ;READ COMP REG
U 1325. 0001.203C.0180.09E8.0000.1326  ;3675  RC[0D]_Q    ;SAVE RECEIVED DATA
U 1326. 0810.0038.0180.0970.0000.1327  ;3676  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 1327. 001D.0000.0180.0800.0010.1328  ;3677  ALU_D-Q,CLK.UBCC
U 1328. 0000.013C.0180.0800.0000.10F2  ;3678  Z?
U 10F2. 0000.003D.0180.0800.0000.11A0  ;3679 =0 ERROR2     ;COMP REG DATA INCORRECT
U 10F3. 0000.003C.0180.0800.0000.10F4  ;3680  NOP     ;DONE
           ;3681 ;
           ;3682 ;
           ;3683 ; SUBTEST 3
           ;3684 ;
U 10F4. 0000.003D.0180.0800.0000.11A2  ;3685 =0 SUBTEST
U 10F5. 0000.003C.0180.0800.0000.10F6  ;3686  NOP
U 10F6. 0000.003D.0180.0800.0000.119D  ;3687 =0 ERLOOP
U 10F7. 0810.0038.0180.0900.0000.1329  ;3688  D_RC[0]     ;GET CPU'S ID
U 1329. 0001.003C.0180.09D8.0000.10F8  ;3689  RC[0B]_D    ;TEST ID <4:0>
U 10F8. 0000.003D.0180.0800.0000.11F2  ;3690 =0 CALL,J/SETMRG
```

```
U 10F9. 0000.003C.7580.3C00.0000.10FA  ;3691  ID[MAINT]_D
U 10FA. 0F00.003D.0180.0800.0000.11DB  ;3692 =0 CALL.J/DCS,D_0
U 10FB. 0001.003C.0180.09E0.0000.10FC  ;3693  RC[0C]_D    ;WRITE DATA TAG
U 10FC. 0000.003D.0180.0800.0000.11F9  ;3694 =0 CALL.J/TGCOMP
U 10FD. 0000.003C.7180.3C00.0000.132A  ;3695  ID[COMP]_D
U 132A. 0000.003C.8D80.0800.0084.732B  ;3696  SC_K[.1F]
U 132B. 0801.001C.0180.0800.0000.132C  ;3697  D_D.ORNOT.MASK    ;SET <31> COMP LOCK BIT
U 132C. 0000.003C.6580.0800.0084.732D  ;3698  SC_K[.10]
U 132D. 0801.001C.0180.0800.0000.132E  ;3699  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 132E. 0001.003C.0180.09F0.0000.132F  ;3700  RC[0E]_D    ;SAVE EXPECTED DATA
U 132F. 0000.003C.8980.0800.0084.7330  ;3701  SC_K[.D]
U 1330. 0003.001C.0180.0AF8.0000.1331  ;3702  R[0F]_ALU.ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 1331. 0F00.003C.0180.0A00.0200.1332  ;3703  VA_R[0],D_0
U 1332. 0000.003C.0180.A800.0000.1333  ;3704  MCT/WRITE.P    ;START SBI CYCLE FOR COMPARE
U 1333. 0000.003C.0180.0800.0000.1334  ;3705  NOP
U 1334. 0000.003C.0180.0800.0000.1335  ;3706  NOP
U 1335. 0000.003C.0180.0800.0000.1336  ;3707  NOP
U 1336. 0000.003C.0180.0800.0000.1337  ;3708  NOP
U 1337. 0000.003C.0180.0800.0000.1338  ;3709  NOP
U 1338. 0000.003C.71F0.2C00.0000.1339  ;3710  Q_ID[COMP]    ;READ COMP REG
U 1339. 0001.203C.0180.09E8.0000.133A  ;3711  RC[0D]_Q    ;SAVE RECEIVED DATA
U 133A. 0810.0038.0180.0970.0000.133B  ;3712  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 133B. 001D.0000.0180.0800.0010.133C  ;3713  ALU_D-Q,CLK.UBCC
U 133C. 0000.013C.0180.0800.0000.10FE  ;3714  Z?
U 10FE. 0000.003D.0180.0800.0000.11A0  ;3715 =0 ERROR2     ;COMP REG DATA INCORRECT
U 10FF. 0000.003C.0180.0800.0000.110A  ;3716  NOP     ;DONE
          ;3717 ;
          ;3718 ; SUBTEST 4
          ;3719 ;
U 110A. 0000.003D.0180.0800.0000.11A2  ;3720 =0  SUBTEST
U 110B. 0000.003C.0180.0800.0000.1110  ;3721  NOP
U 1110. 0000.003D.0180.0800.0000.119D  ;3722 =0 ERLOOP
U 1111. 0810.0038.0180.0900.0000.133D  ;3723  D_RC[0]     ;GET CPU'S ID
U 133D. 0001.003C.0180.09D8.0000.1112  ;3724  RC[0B]_D    ;TEST ID <4:0>
U 1112. 0000.003D.0180.0800.0000.11F2  ;3725 =0 CALL.J/SETMRG
U 1113. 0000.003C.7580.3C00.0000.133E  ;3726  ID[MAINT]_D
U 133E. 0018.0038.D580.09E0.0000.1114  ;3727  RC[0C]_K[.6]    ;ISR TAG
U 1114. 0000.003D.0180.0800.0000.11F9  ;3728 =0 CALL.J/TGCOMP
U 1115. 0000.003C.7180.3C00.0000.133F  ;3729  ID[COMP]_D
U 133F. 0000.003C.8D80.0800.0084.7340  ;3730  SC_K[.1F]
U 1340. 0801.001C.0180.0800.0000.1341  ;3731  D_D.ORNOT.MASK    ;SET <31> COMP LOCK BIT
U 1341. 0000.003C.6580.0800.0084.7342  ;3732  SC_K[.10]
U 1342. 0801.001C.0180.0800.0000.1343  ;3733  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 1343. 0001.003C.0180.09F0.0000.1344  ;3734  RC[0E]_D    ;SAVE EXPECTED DATA
U 1344. 0000.003C.0180.D800.0000.1345  ;3735  MCT/READ.INT.SUM  ;START SBI CYCLE FOR COMPARE
U 1345. 0000.003C.0180.0800.0000.1346  ;3736  NOP
U 1346. 0000.003C.0180.0800.0000.1347  ;3737  NOP
U 1347. 0000.003C.0180.0800.0000.1348  ;3738  NOP
U 1348. 0000.003C.71F0.2C00.0000.1349  ;3739  Q_ID[COMP]    ;READ COMP REG
U 1349. 0001.203C.0180.09E8.0000.134A  ;3740  RC[0D]_Q    ;SAVE RECEIVED DATA
U 134A. 0810.0038.0180.0970.0000.134B  ;3741  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 134B. 001D.0000.0180.0800.0010.134C  ;3742  ALU_D-Q,CLK.UBCC
U 134C. 0000.013C.0180.0800.0000.1116  ;3743  Z?
U 1116. 0000.003D.0180.0800.0000.11A0  ;3744 =0 ERROR2     ;COMP REG DATA INCORRECT
U 1117. 0000.003C.0180.0800.0000.1118  ;3745  NOP     ;DONE
```

```
                ;3746 ;
                ;3747 ; SUBTEST 5
                ;3748 ;
U 1118.  0000.003D.0180.0800.0000.11A2  ;3749 =0 SUBTEST
U 1119.  0000.003C.0180.0800.0000.111A  ;3750  NOP
U 111A.  0000.003D.0180.0800.0000.119D  ;3751 =0 ERLOOP
U 111B.  0810.0038.0180.0900.0000.134D  ;3752  D_RC[0]
U 134D.  0819.0014.0580.0800.0000.134E  ;3753  D_D+K[.1]   ;ADD 1 TO CPU'S ID
U 134E.  0001.003C.0180.0980.0000.134F  ;3754  RC[0]_D
U 134F.  0001.003C.0180.09D8.0000.111C  ;3755  RC[0B]_D   ;TEST ID <4:0>
U 111C.  0000.003D.0180.0800.0000.11F2  ;3756 =0 CALL,J/SETMRG
U 111D.  0000.003C.7580.3C00.0000.1350  ;3757  ID[MAINT]_D
U 1350.  0018.0038.0D80.09E0.0000.111E  ;3758  RC[0C]_K[.3]   ;COMMAND ADDRESS TAG
U 111E.  0000.003D.0180.0800.0000.11F9  ;3759 =0 CALL,J/TGCOMP
U 111F.  0000.003C.7180.3C00.0000.1351  ;3760  ID[COMP]_D
U 1351.  0001.003C.0180.09F0.0000.1352  ;3761  RC[0E]_D   ;SAVE EXPECTED DATA
U 1352.  0000.003C.8980.0800.0084.7353  ;3762  SC_K[.D]
U 1353.  0003.001C.0180.0AF8.0000.1354  ;3763  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 1354.  0F00.003C.0180.0A00.0200.1355  ;3764  VA_R[0],D_0
U 1355.  0000.003C.0180.A800.0000.1356  ;3765  MCT/WRITE.P   ;START SBI CYCLE
U 1356.  0000.003C.0180.0800.0000.1357  ;3766  NOP
U 1357.  0000.003C.0180.0800.0000.1358  ;3767  NOP
U 1358.  0000.003C.0180.0800.0000.1359  ;3768  NOP
U 1359.  0000.003C.71F0.2C00.0000.135A  ;3769  Q_ID[COMP]   ;READ COMP REG
U 135A.  0001.203C.0180.09E8.0000.135B  ;3770  RC[0D]_Q   ;SAVE RECEIVED DATA
U 135B.  0810.0038.0180.0970.0000.135C  ;3771  D_RC[0E]   ;RETRIEVE EXPECTED DATA
U 135C.  001D.0000.0180.0800.0010.135D  ;3772  ALU_D-Q,CLK.UBCC
U 135D.  0000.013C.0180.0800.0000.1120  ;3773  Z?
U 1120.  0000.003D.0180.0800.0000.11A0  ;3774 =0 ERROR2    ;COMP REG DATA INCORRECT
U 1121.  0000.003C.0180.0800.0000.1122  ;3775  NOP    ;DONE
                ;3776 ;
                ;3777 ; SUBTEST 6
                ;3778 ;
U 1122.  0000.003D.0180.0800.0000.11A2  ;3779 =0 SUBTEST
U 1123.  0000.003C.0180.0800.0000.1124  ;3780  NOP
U 1124.  0000.003D.0180.0800.0000.119D  ;3781 =0 ERLOOP
U 1125.  0810.0038.0180.0900.0000.135E  ;3782  D_RC[0]    ;GET CPU'S ID
U 135E.  0001.003C.0180.09D8.0000.1126  ;3783  RC[0B]_D   ;TEST ID <4:0>
U 1126.  0000.003D.0180.0800.0000.11F2  ;3784 =0 CALL,J/SETMRG
U 1127.  0000.003C.7580.3C00.0000.135F  ;3785  ID[MAINT]_D
U 135F.  0018.0038.1980.09E0.0000.1128  ;3786  RC[0C]_K[ZERO]   ;READ DATA TAG
U 1128.  0000.003D.0180.0800.0000.11F9  ;3787 =0 CALL,J/TGCOMP
U 1129.  0000.003C.7180.3C00.0000.1360  ;3788  ID[COMP]_D
U 1360.  0001.003C.0180.09F0.0000.1361  ;3789  RC[0E]_D   ;SAVE EXPECTED DATA IN RCE
U 1361.  0000.003C.8980.0800.0084.7362  ;3790  SC_K[.D]
U 1362.  0003.001C.0180.0AF8.0000.1363  ;3791  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 1363.  0F00.003C.0180.0A00.0200.1364  ;3792  VA_R[0],D_0
U 1364.  0000.003C.0180.A800.0000.1365  ;3793  MCT/WRITE.P   ;START SBI CYCLE
U 1365.  0000.003C.0180.0800.0000.1366  ;3794  NOP
U 1366.  0000.003C.0180.0800.0000.1367  ;3795  NOP
U 1367.  0000.003C.0180.0800.0000.1368  ;3796  NOP
U 1368.  0000.003C.0180.0800.0000.1369  ;3797  NOP
U 1369.  0000.003C.71F0.2C00.0000.136A  ;3798  Q_ID[COMP]   ;READ COMP REG
U 136A.  0001.203C.0180.09E8.0000.136B  ;3799  RC[0D]_Q   ;SAVE RECEIVED DATA
U 136B.  0810.0038.0180.0970.0000.136C  ;3800  D_RC[0E]   ;RETRIEVE EXPECTED DATA
```

```
U 136C, 001D,0000,0180,0800,0010,136D  ;3801  ALU_D-Q,CLK.UBCC
U 136D, 0000,013C,0180,0800,0000,112A  ;3802  Z?
U 112A, 0000,003D,0180,0800,0000,11A0  ;3803 =0 ERROR2     ;COMP REG DATA INCORRECT
U 112B, 0000,003C,0180,0800,0000,112C  ;3804 NODEVS: NOP     ;DONE
            ;3805
```

```
;3806  .PAGE "TEST 195 SILO LOCK ON ID,TAG & FUNCTION COMPARE TEST"
;3807 ;*****************************************************************************
;3808 ;++
;3809 ;
;3810 ;TEST 195  SILO LOCK ON ID,TAG & FUNCTION COMPARE TEST
;3811 ;  FUNCTION COMPARISON TEST
;3812 ;
;3813 ;TEST DESCRIPTION
;3814 ; THIS TEST CHECKS THAT THE SILO LOCKS CORRECTLY WHEN
;3815 ; THE SILO COMPARATOR REGISTER CONDITIONAL LOCK BITS
;3816 ; ARE EQUAL TO 1,1 (LOCK ON ID, TAG AND FUNCTION
;3817 ; COMPARISON). SUBTESTS 1-5 USE THE CPU'S ID AND A COMMAND
;3818 ; ADDRESS TAG (3) FOR COMPARISON. SUBTEST 6 USES THE CPU'S
;3819 ; ID AND AN INVALID TAG OF 7.
;3820 ;
;3821 ; SUBTEST 1 USES A READ MASKED FUNCTION CODE.
;3822 ;   A CPU READ OF I/O SPACE IS EXECUTED.
;3823 ;   THE SILO SHOULD LOCK.
;3824 ; SUBTEST 2 USES A WRITE MASKED FUNCTION CODE.
;3825 ;   A CPU WRITE TO MEMORY IS EXECUTED.
;3826 ;   THE SILO SHOULD LOCK.
;3827 ; SUBTEST 3 USES AN INTERLOCKED READ MASKED
;3828 ;   FUNCTION CODE. A CPU INTERLOCKED READ
;3829 ;   OF MEMORY IS EXECUTED. THE SILO SHOULD
;3830 ;   LOCK.
;3831 ; SUBTEST 4 USES AN EXTENDED READ FUNCTION CODE.
;3832 ;   A CPU READ OF MEMORY IS EXECUTED.
;3833 ;   THE SILO SHOULD LOCK.
;3834 ; SUBTEST 5 USES AN INVALID FUNCTION CODE OF ZERO.
;3835 ;   A CPU WRITE TO MEMORY IS EXECUTED.
;3836 ;   THE SILO SHOULD NOT LOCK.
;3837 ; SUBTEST 6 USES A WRITE MASKED FUNCTION CODE.
;3838 ;   A CPU WRITE TO MEMORY IS EXECUTED.
;3839 ;   THE SILO SHOULD NOT LOCK DUE TO THE INVALID
;3840 ;   TAG WHICH IS USED.
;3841 ;
;3842 ;LOGIC DESCRIPTION
;3843 ;
;3844 ;  N/A
;3845 ;
;3846 ;ERROR DESCIRTION
;3847 ; EXPECTED COMPARATOR REGISTER DATA
;3848 ; RECEIVED COMPARATOR REGISTER DATA
;3849 ; TEST ID <4:0>
;3850 ; TEST TAG <3:0>
;3851 ; TEST FUNCTION <3:0>
;3852 ;
;3853 ;--
;3854 ;*****************************************************************************
;3855 ;
;3856 =0
U 112C, 0000,003D,0180,0800,0000,117C  ;3857 LKITF: NEWTST
U 112D, 0000,003C,0180,0800,0000,112E  ;3858  NOP
U 112E, 0000,003D,0180,0800,0000,1255  ;3859 =0 CALL[CONFIG.MS780E]  ; Make sure that MS780-E/H is
;3860       ; ...properly configured
```

```
U 112F. 0000.003C.0180.0800.0000.1130  ;3861  NOP
U 1130. 0000.003D.0180.0800.0000.11DB  ;3862 =0 CALL.J/DCS
U 1131. 0001.003C.0180.09F8.0000.1132  ;3863  RC[0F]_D
U 1132. 0000.003D.0180.0800.0000.1036  ;3864 =0 CALL.J/CPUCA
U 1133. 0B00.003C.0180.0800.0000.136E  ;3865  D_D.SWAP
U 136E. 0600.003C.0180.0800.0000.136F  ;3866  D_D.RIGHT
U 136F. 0819.0034.8D80.0800.0000.1370  ;3867  D_D.AND.K[.1F]
U 1370. 0001.003C.0180.0980.0000.1134  ;3868  RC[0]_D    ;SAVE CPU'S ID IN RC0 <4:0>
           ;3869 ;
           ;3870 ; SUBTEST 1
           ;3871 ;
U 1134. 0000.003D.0180.0800.0000.11A2  ;3872 =0 SUBTEST
U 1135. 0000.003C.0180.0800.0000.1136  ;3873  NOP
U 1136. 0000.003D.0180.0800.0000.119D  ;3874 =0 ERLOOP
U 1137. 0810.0038.0180.0900.0000.1371  ;3875  D_RC[0]    ;GET CPU'S ID
U 1371. 0001.003C.0180.09E0.0000.1138  ;3876  RC[0C]_D   ;TEST ID <4:0>
U 1138. 0000.003D.0180.0800.0000.11F2  ;3877 =0 CALL.J/SETMRG
U 1139. 0000.003C.7580.3C00.0000.1372  ;3878  ID[MAINT]_D
U 1372. 0018.0038.0D80.09D8.0000.1373  ;3879  RC[0B]_K[.3]   ;TEST TAG=COMMAND ADDRESS
U 1373. 0018.0038.0580.09D0.0000.113A  ;3880  RC[0A]_K[.1]   ;TEST FUNCTION=READ MASKED
U 113A. 0000.003D.0180.0800.0000.11FF  ;3881 =0 CALL.J/FNCOMP   ;SET COMP REG TO LOCK ON ID,
U 113B. 0000.003C.7180.3C00.0000.1374  ;3882  ID[COMP]_D   ;TAG AND FUNCTION COMPARE
U 1374. 0000.003C.8D80.0800.0084.7375  ;3883  SC_K[.1F]
U 1375. 0801.001C.0180.0800.0000.1376  ;3884  D_D.ORNOT.MASK   ;SET SILO LOCK BIT <31>
U 1376. 0000.003C.6580.0800.0084.7377  ;3885  SC_K[.10]
U 1377. 0801.001C.0180.0800.0000.1378  ;3886  D_D.ORNOT.MASK  ;SET COUNT FIELD = F
U 1378. 0001.003C.0180.09F0.0000.1379  ;3887  RC[0E]_D   ;SAVE EXPECTED DATA
U 1379. 0F00.003C.8980.0800.0084.737A  ;3888  SC_K[.D],D_0
U 137A. 0801.001C.0180.0800.0000.137B  ;3889  D_D.ORNOT.MASK
U 137B. 0000.003C.5180.0800.0084.737C  ;3890  SC_K[.1E]
U 137C. 0000.003C.0580.0800.0084.B37D  ;3891  SC_SC-K[.1]
U 137D. 0801.001C.0180.0800.0000.137E  ;3892  D_D.ORNOT.MASK
U 137E. 0001.003C.0180.0800.0200.137F  ;3893  VA_D    ;VA=20002000
U 137F. 0000.003C.8980.0800.0084.7380  ;3894  SC_K[.D]
U 1380. 0003.001C.0180.0AF8.0000.1381  ;3895  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 1381. 0000.003C.0180.C800.0000.1382  ;3896  MCT/READ.P   ;READ MASKED FUNCTION
U 1382. 0000.003C.0180.0800.0000.1383  ;3897  NOP
U 1383. 0000.003C.0180.0800.0000.1384  ;3898  NOP
U 1384. 0000.003C.0180.0800.0000.1385  ;3899  NOP
U 1385. 0000.003C.71F0.2C00.0000.1386  ;3900  Q_ID[COMP]   ;READ COMP REG
U 1386. 0001.203C.0180.09E8.0000.1387  ;3901  RC[0D]_Q   ;SAVE RECEIVED DATA
U 1387. 0810.0038.0180.0970.0000.1388  ;3902  D_RC[0E]   ;RETRIEVE EXPECTED DATA
U 1388. 001D.0000.0180.0800.0010.1389  ;3903  ALU_D-Q,CLK.UBCC
U 1389. 0000.013C.0180.0800.0000.113C  ;3904  Z?
U 113C. 0000.003D.0180.0800.0000.11A0  ;3905 =0 ERROR2    ;COMP REG DATA INCORRECT
U 113D. 0000.003C.0180.0800.0000.113E  ;3906  NOP    ;DONE
           ;3907
           ;3908 ; SUBTEST 2
           ;3909 ;
U 113E. 0000.003D.0180.0800.0000.11A2  ;3910 =0 SUBTEST
U 113F. 0810.0038.0180.0900.0000.138A  ;3911  D_RC[0]    ;GET CPU'S ID
U 138A. 0001.003C.0180.09E0.0000.1140  ;3912  RC[0C]_D   ;TEST ID <4:0>
U 1140. 0000.003D.0180.0800.0000.11F2  ;3913 =0 CALL.J/SETMRG
U 1141. 0000.003C.7580.3C00.0000.138B  ;3914  ID[MAINT]_D
U 138B. 0018.0038.0D80.09D8.0000.138C  ;3915  RC[0B]_K[.3]   ;TEST TAG=COMMAND ADDRESS
```

```
U 138C. 0018.0038.0980.09D0.0000.1142  ;3916  RC[0A]_K[.2]    ;TEST FUNCTION=WRITE MASKED
U 1142. 0000.003D.0180.0800.0000.11FF  ;3917 =0 CALL,J/FNCOMP    ;SET COMP REG TO LOCK ON ID,
U 1143. 0000.003C.7180.3C00.0000.138D  ;3918  ID[COMP]_D    ;TAG AND FUNCTION COMPARE
U 138D. 0000.003C.8D80.0800.0084.738E  ;3919  SC_K[.1F]
U 138E. 0801.001C.0180.0800.0000.138F  ;3920  D_D.ORNOT.MASK    ;SET SILO LOCK BIT <31>
U 138F. 0000.003C.6580.0800.0084.7390  ;3921  SC_K[.10]
U 1390. 0801.001C.0180.0800.0000.1391  ;3922  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 1391. 0001.003C.0180.09F0.0000.1392  ;3923  RC[0E]_D    ;SAVE EXPECTED DATA
U 1392. 0000.003C.8980.0800.0084.7393  ;3924  SC_K[.D]
U 1393. 0003.001C.0180.0AF8.0000.1394  ;3925  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 1394. 0018.0038.1980.0800.0200.1395  ;3926  VA_K[ZERO]
U 1395. 0000.003C.0180.A800.0000.1396  ;3927  MCT/WRITE.P    ;START SBI CYCLE FOR COMPARE
U 1396. 0000.003C.0180.0800.0000.1397  ;3928  NOP
U 1397. 0000.003C.0180.0800.0000.1398  ;3929  NOP
U 1398. 0000.003C.0180.0800.0000.1399  ;3930  NOP
U 1399. 0000.003C.0180.0800.0000.139A  ;3931  NOP
U 139A. 0000.003C.71F0.2C00.0000.139B  ;3932  Q_ID[COMP]    ;READ COMP REG
U 139B. 0001.203C.0180.09E8.0000.139C  ;3933  RC[0D]_Q    ;SAVE RECEIVED DATA
U 139C. 0810.0038.0180.0970.0000.139D  ;3934  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 139D. 001D.0000.0180.0800.0010.139E  ;3935  ALU_D-Q,CLK.UBCC
U 139E. 0000.013C.0180.0800.0000.1144  ;3936  Z?
U 1144. 0000.003D.0180.0800.0000.11A0  ;3937 =0 ERROR2    ;COMP REG DATA INCORRECT
U 1145. 0000.003C.0180.0800.0000.1146  ;3938  NOP    ;DONE
          ;3939 ;
          ;3940 ; SUBTEST 3
          ;3941 ;
U 1146. 0000.003D.0180.0800.0000.11A2  ;3942 =0 SUBTEST
U 1147. 0000.003C.0180.0800.0000.1148  ;3943  NOP
U 1148. 0000.003D.0180.0800.0000.119D  ;3944 =0 ERLOOP
U 1149. 0810.0038.0180.0900.0000.139F  ;3945  D_RC[0]    ;GET CPU'S ID
U 139F. 0001.003C.0180.09E0.0000.114A  ;3946  RC[0C]_D    ;TEST ID <4:0>
U 114A. 0000.003D.0180.0800.0000.11F2  ;3947 =0 CALL,J/SETMRG
U 114B. 0000.003C.7580.3C00.0000.13A0  ;3948  ID[MAINT]_D
U 13A0. 0018.0038.0D80.09D8.0000.13A1  ;3949  RC[0B]_K[.3]    ;TEST TAG COMMAND ADDRESS
U 13A1. 0018.0038.1180.09D0.0000.114C  ;3950  RC[0A]_K[.4]    ;TEST FUNCTION INT. READ MASKED
U 114C. 0000.003D.0180.0800.0000.11FF  ;3951 =0 CALL,J/FNCOMP    ;SET COMP REG TO LOCK ON ID
U 114D. 0000.003C.7180.3C00.0000.13A2  ;3952  ID[COMP]_D    ;TAG AND FUNCTION COMPARE
U 13A2. 0000.003C.8D80.0800.0084.73A3  ;3953  SC_K[.1F]
U 13A3. 0801.001C.0180.0800.0000.13A4  ;3954  D_D.ORNOT.MASK    ;SET SILO COMP BIT <31>
U 13A4. 0000.003C.6580.0800.0084.73A5  ;3955  SC_K[.10]
U 13A5. 0801.001C.0180.0800.0000.13A6  ;3956  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 13A6. 0001.003C.0180.09F0.0000.13A7  ;3957  RC[0E]_D    ;SAVE EXPECTED DATA
U 13A7. 0000.003C.8980.0800.0084.73A8  ;3958  SC_K[.D]
U 13A8. 0003.001C.0180.0AF8.0000.13A9  ;3959  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 13A9. 0018.0038.1980.0800.0200.13AA  ;3960  VA_K[ZERO]
U 13AA. 0000.003C.0180.E800.0000.13AB  ;3961  MCT/LOCKREAD.P    ;START SBI CYCLE FOR COMPARE
U 13AB. 0000.003C.0180.0800.0000.13AC  ;3962  NOP
U 13AC. 0000.003C.0180.0800.0000.13AD  ;3963  NOP
U 13AD. 0000.003C.0180.0800.0000.13AE  ;3964  NOP
U 13AE. 0000.003C.71F0.2C00.0000.13AF  ;3965  Q_ID[COMP]    ;READ COMP REG
U 13AF. 0001.203C.0180.09E8.0000.13B0  ;3966  RC[0D]_Q    ;SAVE RECEIVED DATA
U 13B0. 0018.0038.1980.0800.0200.13B1  ;3967  VA_K[ZERO]
U 13B1. 0000.003C.0180.B800.0000.13B2  ;3968  MCT/LOCKWRITE.P    ;CLEAR INTERLOCK FF
U 13B2. 0810.0038.0180.0970.0000.13B3  ;3969  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 13B3. 001D.0000.0180.0800.0010.13B4  ;3970  ALU_D-Q,CLK.UBCC
```

```
U 13B4. 0000.013C.0180.0800.0000.114E  ;3971  Z?
U 114E. 0000.003D.0180.0800.0000.11A0  ;3972 =0 ERROR2     ;COMP REG DATA INCORRECT
U 114F. 0000.003C.0180.0800.0000.1150  ;3973  NOP     ;DONE
          ;3974 ;
          ;3975 ; SUBTEST 4
          ;3976 ;
U 1150. 0000.003D.0180.0800.0000.11A2  ;3977 =0 SUBTEST
U 1151. 0000.003C.0180.0800.0000.1152  ;3978  NOP
U 1152. 0000.003D.0180.0800.0000.119D  ;3979 =0 ERLOOP
U 1153. 0810.0038.0180.0900.0000.13B5  ;3980  D_RC[0]     ;GET CPU'S ID
U 13B5. 0001.003C.0180.09E0.0000.1156  ;3981  RC[0C]_D     ;TEST ID <4:0>
U 1156. 0000.003D.0180.0800.0000.11F2  ;3982 =0 CALL,J/SETMRG
U 1157. 0000.003C.7580.3C00.0000.13B6  ;3983  ID[MAINT]_D
U 13B6. 0018.0038.0D80.09D8.0000.13B7  ;3984  RC[0B]_K[.3]    ;TEST TAG COMMAND ADDRESS
U 13B7. 0018.0038.0180.09D0.0000.1158  ;3985  RC[0A]_K[.8]    ;TEST FUNCTION EXTENDED READ
U 1158. 0000.003D.0180.0800.0000.11FF  ;3986 =0 CALL,J/FNCOMP    ;SET COMP REG TO LOCK ON ID
U 1159. 0000.003C.7180.3C00.0000.13B8  ;3987  ID[COMP]_D    ;TAG AND FUNCTION COMPARE
U 13B8. 0000.003C.8D80.0800.0084.73B9  ;3988  SC_K[.1F]
U 13B9. 0801.001C.0180.0800.0000.13BA  ;3989  D_D.ORNOT.MASK    ;SET SILO LOCK BIT <31>
U 13BA. 0000.003C.6580.0800.0084.73BB  ;3990  SC_K[.10]
U 13BB. 0801.001C.0180.0800.0000.13BC  ;3991  D_D.ORNOT.MASK    ;SET COUNT FIELD = F
U 13BC. 0001.003C.0180.09F0.0000.13BD  ;3992  RC[0E]_D    ;SAVE EXPECTED DATA
U 13BD. 0000.003C.8980.0800.0084.73BE  ;3993  SC_K[.D]
U 13BE. 0003.001C.0180.0AF8.0000.13BF  ;3994  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 13BF. 0018.0038.1980.0800.0200.13C0  ;3995  VA_K[ZERO]
U 13C0. 0000.003C.0180.C800.0000.13C1  ;3996  MCT/READ.P    ;START SBI CYCLE FOR COMPARE
U 13C1. 0000.003C.0180.0800.0000.13C2  ;3997  NOP
U 13C2. 0000.003C.0180.0800.0000.13C3  ;3998  NOP
U 13C3. 0000.003C.0180.0800.0000.13C4  ;3999  NOP
U 13C4. 0000.003C.71F0.2C00.0000.13C5  ;4000  Q_ID[COMP]    ;READ COMP REG
U 13C5. 0001.203C.0180.09E8.0000.13C6  ;4001  RC[0D]_Q    ;SAVE RECEIVED DATA
U 13C6. 0810.0038.0180.0970.0000.13C7  ;4002  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 13C7. 001D.0000.0180.0800.0010.13C8  ;4003  ALU_D-Q,CLK.UBCC
U 13C8. 0000.013C.0180.0800.0000.115A  ;4004  Z?
U 115A. 0000.003D.0180.0800.0000.11A0  ;4005 =0 ERROR2     ;COMP REG DATA INCORRECT
U 115B. 0000.003C.0180.0800.0000.115C  ;4006  NOP     ;DONE
          ;4007 ;
          ;4008 ; SUBTEST 5
          ;4009 ;
U 115C. 0000.003D.0180.0800.0000.11A2  ;4010 =0 SUBTEST
U 115D. 0000.003C.0180.0800.0000.115E  ;4011  NOP
U 115E. 0000.003D.0180.0800.0000.119D  ;4012 =0 ERLOOP
U 115F. 0810.0038.0180.0900.0000.13C9  ;4013  D_RC[0]     ;GET CPU'S ID
U 13C9. 0001.003C.0180.09E0.0000.1160  ;4014  RC[0C]_D     ;TEST ID <4:0>
U 1160. 0000.003D.0180.0800.0000.11F2  ;4015 =0 CALL,J/SETMRG
U 1161. 0000.003C.7580.3C00.0000.13CA  ;4016  ID[MAINT]_D
U 13CA. 0018.0038.0D80.09D8.0000.13CB  ;4017  RC[0B]_K[.3]    ;TEST TAG COMMAND ADDRESS
U 13CB. 0018.0038.1980.09D0.0000.1162  ;4018  RC[0A]_K[ZERO]    ;TEST FUNCTION NON-EXISTENT
U 1162. 0000.003D.0180.0800.0000.11FF  ;4019 =0 CALL,J/FNCOMP    ;SET COMP REG TO LOCK ON ID
U 1163. 0000.003C.7180.3C00.0000.13CC  ;4020  ID[COMP]_D    ;TAG AND FUNCTION COMPARE
U 13CC. 0001.003C.0180.09F0.0000.13CD  ;4021  RC[0E]_D    ;SAVE EXPECTED DATA
U 13CD. 0000.003C.8980.0800.0084.73CE  ;4022  SC_K[.D]
U 13CE. 0003.001C.0180.0AF8.0000.13CF  ;4023  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 13CF. 0018.0038.1980.0800.0200.13D0  ;4024  VA_K[ZERO]
U 13D0. 0000.003C.0180.A800.0000.13D1  ;4025  MCT/WRITE.P    ;START SBI CYCLE
```

```
U 13D1. 0000.003C.0180.0800.0000.13D2  ;4026  NOP
U 13D2. 0000.003C.0180.0800.0000.13D3  ;4027  NOP
U 13D3. 0000.003C.0180.0800.0000.13D4  ;4028  NOP
U 13D4. 0000.003C.71F0.2C00.0000.13D5  ;4029  Q_ID[COMP]    ;READ COMP REG
U 13D5. 0001.203C.0180.09E8.0000.13D6  ;4030  RC[0D]_Q    ;SAVE RECEIVED DATA
U 13D6. 0810.0038.0180.0970.0000.13D7  ;4031  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 13D7. 001D.0000.0180.0800.0010.13D8  ;4032  ALU_D-Q,CLK.UBCC
U 13D8. 0000.013C.0180.0800.0000.1164  ;4033  Z?
U 1164. 0000.003D.0180.0800.0000.11A0  ;4034 =0 ERROR2     ;COMP REG DATA INCORRECT
U 1165. 0000.003C.0180.0800.0000.1166  ;4035  NOP     ;DONE
          ;4036 ;
          ;4037 ; SUBTEST 6
          ;4038 ;
U 1166. 0000.003D.0180.0800.0000.11A2  ;4039 =0 SUBTEST
U 1167. 0000.003C.0180.0800.0000.1168  ;4040  NOP
U 1168. 0000.003D.0180.0800.0000.119D  ;4041 =0 ERLOOP
U 1169. 0810.0038.0180.0900.0000.13D9  ;4042  D_RC[0]     ;GET CPU'S ID
U 13D9. 0001.003C.0180.09E0.0000.116A  ;4043  RC[0C]_D    ;TEST ID <4:0>
U 116A. 0000.003D.0180.0800.0000.11F2  ;4044 =0 CALL,J/SETMRG
U 116B. 0000.003C.7580.3C00.0000.13DA  ;4045  ID[MAINT]_D
U 13DA. 0018.0038.5D80.09D8.0000.13DB  ;4046  RC[0B]_K[.7]    ;TEST TAG <3:0>
U 13DB. 0018.0038.0980.09D0.0000.116C  ;4047  RC[0A]_K[.2]    ;TEST FUNCTION WRITE MASKED
U 116C. 0000.003D.0180.0800.0000.11FF  ;4048 =0 CALL,J/FNCOMP
U 116D. 0000.003C.7D80.0800.0084.73DC  ;4049  SC_K[.18]
U 13DC. 0000.003C.0980.0800.0084.B3DD  ;4050  SC_SC-K[.2]
U 13DD. 0801.001C.0180.0800.0000.13DE  ;4051  D_D.ORNOT.MASK    ;SET TAG FIELD = 7
U 13DE. 0000.003C.7180.3C00.0000.13DF  ;4052  ID[COMP]_D
U 13DF. 0001.003C.0180.09F0.0000.13E0  ;4053  RC[0E]_D    ;SAVE EXPECTED DATA
U 13E0. 0000.003C.8980.0800.0084.73E1  ;4054  SC_K[.D]
U 13E1. 0003.001C.0180.0AF8.0000.13E2  ;4055  R[0F]_ALU,ALU_NOT.MASK  ;SET T.O. TRAP FLAG
U 13E2. 0018.0038.1980.0800.0200.13E3  ;4056  VA_K[ZERO]
U 13E3. 0000.003C.0180.A800.0000.13E4  ;4057  MCT/WRITE.P    ;START SBI CYCLE
U 13E4. 0000.003C.0180.0800.0000.13E5  ;4058  NOP
U 13E5. 0000.003C.0180.0800.0000.13E6  ;4059  NOP
U 13E6. 0000.003C.0180.0800.0000.13E7  ;4060  NOP
U 13E7. 0000.003C.71F0.2C00.0000.13E8  ;4061  Q_ID[COMP]    ;READ COMP REG
U 13E8. 0001.203C.0180.09E8.0000.13E9  ;4062  RC[0D]_Q    ;SAVE RECEIVED DATA
U 13E9. 0810.0038.0180.0970.0000.13EA  ;4063  D_RC[0E]    ;RETRIEVE EXPECTED DATA
U 13EA. 001D.0000.0180.0800.0010.13EB  ;4064  ALU_D-Q,CLK.UBCC
U 13EB. 0000.013C.0180.0800.0000.116E  ;4065  Z?
U 116E. 0000.003D.0180.0800.0000.11A0  ;4066 =0 ERROR2     ;COMP REG DATA INCORRECT
U 116F. 0000.003C.0180.0800.0000.1170  ;4067  NOP     ;DONE
          ;4068
```

```
            ;4069  .PAGE "TEST 196 RESERVED"
            ;4070 =0
U 1170, 0000,003D,0180,0800,0000,117C  ;4071 T196: NEWTST
U 1171, 0000,003C,0180,0800,0000,1172  ;4072  NOP
            ;4073
```

```
            ;4074  .PAGE "TEST 197 RESERVED"
            ;4075 =0
U 1172, 0000,003D,0180,0800,0000,117C  ;4076 T197: NEWTST
U 1173, 0000,003C,0180,0800,0000,1174  ;4077  NOP
            ;4078
```

```
            ;4079  .PAGE "TEST 198 RESERVED"
            ;4080 =0
U 1174. 0000.003D.0180.0800.0000.117C  ;4081 T198: NEWTST
U 1175. 0000.003C.0180.0800.0000.1176  ;4082  NOP
            ;4083
```

```
            ;4084  .PAGE "DUMMY NEWTST"
            ;4085 =0
U 1176. 0000,003D,0180,0800,0000,117C  ;4086 DUMMY: NEWTST
            ;4087
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 2865= 2866= 2867= 2869=
U 1008  2640= 2641= 2642= 2643= 2644= 2645= 2646= 2647=
U 1010  2914= 2915= 2916= 2918= 3058= 3059= 3060= 1875
U 1018  3065= 3066= 3067= 1876 3071= 3072= 3073= 1877
U 1020  3077= 3080= 3082= 3083= 1898= 1899= 1951= 1953=
U 1028  1964= 1965= 2105= 2106= 2139= 2140= 2259= 2260=
U 1030  2282= 2283= 2287= 2288= 2292= 2293= 2314= 2315=
U 1038  2316= 2317= 2318= 2320= 2321= 2322= 2323= 2324=
U 1040  2346= 2347= 2370= 2371= 2373= 2374= 2380= 2381=
U 1048  2486= 2487= 2507= 2508= 2551= 2552= 2087: 1878
U 1050  2576= 2577= 2578= 2579= 2619= 2620= 2628= 2629=
U 1058  2630= 2632= 2122: 1879 2635= 2636= 2156: 1880
U 1060  2677= 2678= 2679= 2680= 2687= 2688= 2694= 2696=
U 1068  2699= 2700= 2719= 2720= 2722= 2723= 2820= 2821=
U 1070  2949= 2950= 2953= 2954= 2965= 2966= 2969= 2970=
U 1078  2997= 2998= 3009= 3010= 3011= 3012= 3162= 3163=
U 1080  3164= 3167= 3179= 3180= 3197= 3212= 3217= 3219=
U 1088  3253= 3254= 3255= 3256= 3257= 3258= 3260= 3261=
U 1090  3283= 3284= 3285= 3286= 3292= 3293= 3295= 3296=
U 1098  3300= 3301= 3312= 3313= 3374= 3375= 3376= 3378=
U 10A0  3382= 3383= 3384= 3385= 3390= 3391= 3393= 3394=
U 10A8  3395= 3396= 3397= 3398= 3409= 3410= 3415= 3416=
U 10B0  3417= 3418= 3428= 3429= 3430= 3431= 3433= 3434=
U 10B8  3435= 3436= 3448= 3449= 3454= 3455= 3458= 3459=
U 10C0  3460= 3461= 3471= 3472= 3473= 3474= 3482= 3483=
U 10C8  3484= 3485= 3489= 3490= 3504= 3505= 3515= 3516=
U 10D0  3521= 3522= 3578= 3579= 3580= 3582= 3585= 3586=
U 10D8  3587= 3588= 3597= 3598= 3607= 3608= 3610= 3611=
U 10E0  3618= 3619= 3620= 3621= 3623= 3624= 3626= 3627=
U 10E8  3646= 3647= 3651= 3652= 3653= 3654= 3656= 3657=
U 10F0  3659= 3660= 3679= 3680= 3685= 3686= 3687= 3688=
U 10F8  3690= 3691= 3692= 3693= 3694= 3695= 3715= 3716=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1900 3720= 3721= 1870: 1871: 1872: 1873:
U 1110  3722= 3723= 3725= 3726= 3728= 3729= 3744= 3745=
U 1118  3749= 3750= 3751= 3752= 3756= 3757= 3759= 3760=
U 1120  3774= 3775= 3779= 3780= 3781= 3782= 3784= 3785=
U 1128  3787= 3788= 3803= 3804= 3857= 3858= 3859= 3861=
U 1130  3862= 3863= 3864= 3865= 3872= 3873= 3874= 3875=
U 1138  3877= 3878= 3881= 3882= 3905= 3906= 3910= 3911=
U 1140  3913= 3914= 3917= 3918= 3937= 3938= 3942= 3943=
U 1148  3944= 3945= 3947= 3948= 3951= 3952= 3972= 3973=
U 1150  3977= 3978= 3979= 3980= 1901 2186: 3982= 3983=
U 1158  3986= 3987= 4005= 4006= 4010= 4011= 4012= 4013=
U 1160  4015= 4016= 4019= 4020= 4034= 4035= 4039= 4040=
U 1168  4041= 4042= 4044= 4045= 4048= 4049= 4066= 4067=
U 1170  4071= 4072= 4076= 4077= 4081= 4082= 4086= 1902
U 1178  1903 1904 1905 1906 1940 1941 1942 1943
U 1180  1944 1945 1946 1947 1948 1949 1950 1954
U 1188  1955 1956 1957 1958 1959 1960 1961 1962
U 1190  1963 1984 1985 1986 1987 2006 2007 2008
U 1198  2009 2036 2037 2038 2061 2073 2085 2086
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2120 2121 2203 2204 2219 2220 2221 2234
U 11A8  2235 2236 2261 2262 2263 2264 2279 2280
U 11B0  2281 2284 2285 2286 2289 2290 2291 2325
U 11B8  2326 2327 2341 2342 2343 2344 2345 2372
U 11C0  2375 2376 2377 2378 2379 2382 2396 2397
U 11C8  2398 2399 2400 2401 2402 2403 2404 2419
U 11D0  2420 2421 2422 2423 2424 2425 2448 2449
U 11D8  2450 2451 2452 2453 2454 2455 2456 2457
U 11E0  2458 2459 2460 2461 2462 2463 2465 2466
U 11E8  2467 2468 2470 2485 2488 2489 2490 2491
U 11F0  2506 2509 2527 2528 2529 2530 2531 2532
U 11F8  2533 2550 2553 2554 2555 2556 2557 2574
U 1200  2575 2580 2581 2621 2622 2623 2624 2633
U 1208  2634 2638 2639 2652 2653 2655 2656 2657
U 1210  2659 2664 2665 2666 2668 2673 2675 2676
U 1218  2684 2685 2689 2690 2691 2692 2693 2701
U 1220  2702 2703 2705 2706 2707 2708 2709 2710
U 1228  2715 2717 2718 2721 2746 2747 2748 2749
U 1230  2778 2779 2781 2782 2783 2809 2810 2811
U 1238  2812 2813 2814 2815 2816 2818 2819 2945
U 1240  2947 2948 2952 2956 2957 2958 2959 2961
U 1248  2962 2963 2964 2967 2968 2996 3001 3002
U 1250  3004 3005 3006 3007 3008 3056 3057 3084
U 1258  3169 3170 3172 3173 3174 3176 3181 3183
U 1260  3184 3186 3188 3190 3192 3215 3216 3259
U 1268  3262 3263 3264 3265 3266 3267 3268 3269
U 1270  3270 3271 3272 3273 3274 3275 3276 3277
U 1278  3278 3279 3280 3281 3282 3287 3288 3289
U 1280  3290 3291 3294 3297 3298 3299 3302 3303
U 1288  3304 3305 3306 3307 3308 3309 3310 3311
U 1290  3314 3315 3316 3386 3387 3388 3389 3392
U 1298  3399 3400 3401 3402 3403 3404 3405 3406
U 12A0  3407 3408 3419 3420 3421 3422 3423 3424
U 12A8  3425 3426 2187: 3427 3432 3437 3438 3439
U 12B0  3440 3441 3442 3443 3444 3445 3446 3447
U 12B8  3450 3451 3452 3453 3462 3463 3464 3465
U 12C0  3466 3467 3468 3469 3470 3475 3476 3477
U 12C8  3478 3479 3480 3481 3486 3487 3488 3491
U 12D0  3492 3493 3494 3495 3496 3497 3498 3499
U 12D8  3500 3501 3502 3503 3506 3507 3508 3509
U 12E0  3510 3511 3512 3513 3514 3517 3518 3519
U 12E8  3520 3523 3524 3583 3584 3589 3590 3591
U 12F0  3592 3593 3594 3595 3596 3599 3600 3601
U 12F8  3602 3603 3604 3605 3606 3609 3612 3613
U 1300  3614 3622 3625 3628 3629 3630 3631 3632
U 1308  3633 3634 3635 3636 3637 3638 3639 3640
U 1310  3641 3642 3643 3644 3645 3655 3658 3661
U 1318  3662 3663 3664 3665 3666 3667 3668 3669
U 1320  3670 3671 3672 3673 3674 3675 3676 3677
U 1328  3678 3689 3696 3697 3698 3699 3700 3701
U 1330  3702 3703 3704 3705 3706 3707 3708 3709
U 1338  3710 3711 3712 3713 3714 3724 3727 3730
U 1340  3731 3732 3733 3734 3735 3736 3737 3738
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3739 3740 3741 3742 3743 3753 3754 3755
U 1350  3758 3761 3762 3763 3764 3765 3766 3767
U 1358  3768 3769 3770 3771 3772 3773 3783 3786
U 1360  3789 3790 3791 3792 3793 3794 3795 3796
U 1368  3797 3798 3799 3800 3801 3802 3866 3867
U 1370  3868 3876 3879 3880 3883 3884 3885 3886
U 1378  3887 3888 3889 3890 3891 3892 3893 3894
U 1380  3895 3896 3897 3898 3899 3900 3901 3902
U 1388  3903 3904 3912 3915 3916 3919 3920 3921
U 1390  3922 3923 3924 3925 3926 3927 3928 3929
U 1398  3930 3931 3932 3933 3934 3935 3936 3946
U 13A0  3949 3950 3953 3954 3955 3956 3957 3958
U 13A8  3959 3960 3961 3962 3963 3964 3965 3966
U 13B0  3967 3968 3969 3970 3971 3981 3984 3985
U 13B8  3988 3989 3990 3991 3992 3993 3994 3995
U 13C0  3996 3997 3998 3999 4000 4001 4002 4003
U 13C8  4004 4014 4017 4018 4021 4022 4023 4024
U 13D0  4025 4026 4027 4028 4029 4030 4031 4032
U 13D8  4033 4043 4046 4047 4050 4051 4052 4053
U 13E0  4054 4055 4056 4057 4058 4059 4060 4061
U 13E8  4062 4063 4064 4065
U 13F0  2170: 2171: 2172: 2173: 2174: 2175: 2176: 2177:
U 13F8  2178: 2179: 2180: 2181: 2182: 2183: 2184: 2185:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR41	1020
Used	1020
Remaining	

Total microwords used in memory U: 1020
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR41.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR42.MCR MICRO2 1P(00) 29-JUL-85 15 SEQ 0438

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1847 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1914 Micro Monitor Interface Routines
; 1915   Newtest Routine
; 1972   ERLOOP ROUTINE
; 1976   ERROR1 ROUTINE
; 1982   ERROR1 WITH PARAMETER
; 2004   ERROR2 ROUTINE
; 2010   ERROR 2 WITH PARAMETER
; 2031   CALLCON ROUTINE
; 2037   SCRATCH PAD LOCATIONS
; 2061   SUBTEST ROUTINE
; 2072   SUBTEST 1 ROUTINE
; 2083   SETRCF ROUTINE
; 2094   ENBLFLT ROUTINE
; 2111   NOINTR ROUTINE
; 2156   GENREATE CONSTANT ROUTINE (DCE)
; 2182   DELAY ROUTINE
; 2186 DIAGNOSTIC SPECIFIC ROUTINES
; 2188   CHECK UTRAP
; 2220   Set Trap Mask
; 2249   Disable Cache Routine
; 2271   Initialize the SBI
; 2300   FIND MS780-E MEMORY
; 2442   Generate IO Address
; 2469   Set Starting Address
; 2503   ENABLE NEXT MS780-E
; 2554   Select Controller
; 2592   SELECT LOWER CONTROLLER
; 2640   SELECT UPPER CONTROLLER
; 2689   CLRSBI ROUTINE
; 2712   CPUCA ROUTINE
; 2742   SILCLR ROUTINE
; 2758   SLOLCK ROUTINE
; 2790   CLCPTO ROUTINE
; 2799   SETMRG ROUTINE
; 2819   INIT ROUTINE
; 2851   CHECK FOR INTERRUPT VECTORS
; 2898   SETPSL ROUTINE
; 2912   CLRTIM ROUTINE
; 2926   CLRFLT ROUTINE
; 2940   CLRECC ROUTINE
; 2954   ENBECC ROUTINE
; 2967   Wait Loop Routine
; 2998   RESET SUBTEST NUMBER
; 3020   WAIT REFRESH SUBROUTINE
; 3081 TEST 199 SILO INTEGRITY TEST
; 3189 TEST 19A CHECK SBI COMMANDS AND STALL USING SILO TEST 1
; 3892 TEST 19B RESERVED
; 3897 TEST 19C RESERVED

```
: 3902 DUMMY NEWTST
```

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR42
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; 2.1 Joe Zagarella 11-jul-1985
;1836 ;  With the ms780-e/h configured with only the upper
;1837 ;  controller, test 19a would fail because it only
;1838 ;  checked for the lower controller. Added code at
;1839 ;  label "try.upper" to check for the upper controller.
;1840 ;
;1841 ;
;1842 ;
;1843 ;*****************************************************************************
;1844
```

```
                ;1845  .PAGE
                ;1846
                ;1847  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
                ;1848 ;*
                ;1849 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1850 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
                ;1851 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1852 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1853 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1854 ;
                ;1855 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1856 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1857 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1858 ; R[0F] AND DO A RETURN0.
                ;1859 ;
                ;1860 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1861 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1862 ; TO THE CONSOLE.
                ;1863 ;
                ;1864 ;-
U 1100, 0000,003C,1980,0800,0084,7003  ;1865 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1866 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000,003C,0980,0800,0084,7001  ;1867 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103, 0000,003C,0D80,0800,0084,7001  ;1868 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104, 0000,003C,1180,0800,0084,7001  ;1869 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105, 0000,003C,D580,0800,0084,7001  ;1870 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106, 0000,003C,D980,0800,0084,7001  ;1871 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000,003C,5D80,0800,0084,7001  ;1872 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108, 0000,003C,0180,0800,0084,7001  ;1873 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C, 0000,003C,8580,0800,0084,7001  ;1874 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D, 0000,003C,8980,0800,0084,7001  ;1875 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E, 0000,003C,6580,0800,0084,7001  ;1876 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F, 0000,003C,6180,0800,0084,7003  ;1877 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000,003C,81F0,2C00,0000,1017  ;1878 TRPH0: Q_ID[USTACK]
U 1017, 0C00,003C,01E0,0800,0000,104F  ;1879  Q_D,D_Q
U 104F, 0000,003C,8180,3C00,0000,105B  ;1880  ID[USTACK]_D
U 105B, 0C00,003C,01E0,0800,0000,105F  ;1881  Q_D,D_Q
U 105F, 0000,003C,01C0,0A78,0000,1109  ;1882  Q_R[0F]
U 1109, 0001,2024,0180,0800,0010,113D  ;1883  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 113D, 0000,013C,0180,0800,0000,1002  ;1884  Z?     ; BRANCH IF NO
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1885 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1886 ;*
                ;1887 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1888 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1889 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1890 ;
                ;1891 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1892 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1893 ;       FAULT STATUS REGISTER
                ;1894 ;       SBI ERROR REGISTER
                ;1895 ;       TIMEOUT ADDRESS REGISTER
                ;1896 ;       MAINTENANCE REGISTER
                ;1897 ;       CACHE PARITY REGISTER
                ;1898 ;       TB ERROR REGISTER 1
                ;1899 ;       TB ERROR REGISTER 0
```

```
            ;1900 ;-
U 1003, 0018,0038,1D80,09E8,0000,1038  ;1901 TRPH1: RC[0D]_SC
U 1038, 0000,003D,D980,0800,0084,7164  ;1902 =0 SETRCF[.9]
U 1039, 0000,003C,49F0,2C00,0000,113E  ;1903  Q_ID[TBER0]
U 113E, 0001,203C,4DF0,2DB0,0000,113F  ;1904  RC[6]_Q,Q_ID[TBER1]
U 113F, 0001,203C,65F0,2DB8,0000,1140  ;1905  RC[7]_Q,Q_ID[SBI.ERR]
U 1140, 0001,203C,6DF0,2DD8,0000,1141  ;1906  RC[0B]_Q,Q_ID[FAULT]
U 1141, 0001,203C,75F0,2DE0,0000,1142  ;1907  RC[0C]_Q,Q_ID[MAINT]
U 1142, 0001,203C,79F0,2DC8,0000,1143  ;1908  RC[9]_Q,Q_ID[PARITY]
U 1143, 0001,203C,69F0,2DC0,0000,1144  ;1909  RC[8]_Q,Q_ID[TIME.ADDR]
U 1144, 0001,203C,81F0,2DD0,0000,1000  ;1910  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001,203D,0180,09F0,0000,115C  ;1911 1000: RC[0E]_Q,ERROR1
            ;1912
```

```
                ;1913  .PAGE
                ;1914  .TOC " Micro Monitor Interface Routines"
                ;1915  .TOC "   Newtest Routine"
                ;1916 ;**
                ;1917 ; FUNCTIONAL DESCRIPTION:
                ;1918 ;
                ;1919 ; This routine Initializes the following registers:
                ;1920 ;  PSL    to 1F
                ;1921 ;  CES    to 0
                ;1922 ;  ACC.CS    to disable accellerator
                ;1923 ;  SIR    to 0
                ;1924 ;  CLK.CS    to stop interval timer
                ;1925 ;  TBER0    to disable the TB
                ;1926 ;  SBI.MAINT to     0
                ;1927 ;  SBI.ERR    to 0
                ;1928 ;  FAULT    to 0
                ;1929 ;  COMP    to 0
                ;1930 ;  RC[0E]    to  0
                ;1931 ;  R[0F]    to 0
                ;1932 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1933 ; A SCOPE call is then made to the Monitor.
                ;1934 ;
                ;1935 ; CALLING CONSTRAINT:
                ;1936 ;
                ;1937 ; =0
                ;1938 ;
                ;1939 ; SIDE EFFECTS:
                ;1940 ;
                ;1941 ; D Reg is destroyed
                ;1942 ; SC is destroyed
                ;1943 ;--
U 1145, 0000,003C,65F8,0800,0084,7146   ;1944 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1146, 0818,0038,8D80,0800,0000,1147   ;1945  d_k[.1f]  ; D=1F
U 1147, 0D00,003C,0180,0800,0000,1148   ;1946  d_dai.sc  ; D=001F0000
U 1148, 0000,003C,3D80,3C00,0000,1149   ;1947  id[psl]_d  ; psl = 001F0000
U 1149, 0818,0038,0580,0800,0000,114A   ;1948  d_k[.1]  ; Clear the CES register
U 114A, 0D00,003C,0180,0800,0000,114B   ;1949  d_dai.sc  ; D = 00010000
U 114B, 0F00,003C,3180,3C00,0000,114C   ;1950  id[ces]_d,d_0  ; ces = 00010000
U 114C, 0000,003C,5D80,3C00,0000,114D   ;1951  id[acc.cs]_d  ; acc.cs = 0
U 114D, 0000,003C,3980,3C00,0000,114E   ;1952  id[sir]_d  ; sir = 0
U 114E, 0000,003C,2980,3C00,0000,114F   ;1953  id[clk.cs]_d  ; clk.cs = 0
U 114F, 0000,003C,4980,3C00,0000,103A   ;1954  id[tber0]_d  ; tber0 = 0
U 103A, 0000,003D,0180,0800,0000,11DB   ;1955 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1956  intrpt.strobe,  ; Strobe interrupts
U 103B, 0818,0038,C180,0800,4000,1150   ;1957      d_k[.ffff]  ; Setup to clear SBI error register and
U 1150, 0801,0028,6580,3C00,0000,1151   ;1958  id[sbi.err]_d,d_not.d ; and fault register
U 1151, 0F00,003C,6D80,3C00,0000,1152   ;1959  id[fault]_d,d_0  ; ...
U 1152, 0000,003C,6D80,3C00,0000,1153   ;1960  id[fault]_d  ; fault = 0
U 1153, 0000,003C,7580,3C00,0000,1154   ;1961  id[maint]_d  ; MAINT = 0
U 1154, 0000,003C,6580,3C00,0000,1156   ;1962  id[sbi.err]_d  ; sbi error reg = 0
U 1156, 0000,003C,7180,3C00,0000,1157   ;1963  id[comp]_d  ; comp reg = 0
U 1157, 0000,003C,E580,3C00,0000,1158   ;1964  id[39]_d  ; Clear code for subroutine START.TEST
U 1158, 0001,003C,0180,09F0,0000,1159   ;1965  rc[0e]_d  ;
U 1159, 0001,003C,0180,0AF8,0000,115A   ;1966  r[0f]_d  ; Clear expected trap mask
U 115A, 0001,003C,0180,09F8,0000,103C   ;1967  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 103C, 0000,003D,0180,0800,0000,1189  ;1968 =0 call[disable.cache] ; Disable the cache
U 103D, 0F03,003C,0180,09E8,0000,105E  ;1969  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
            ;1970
```

```
            ;1971   .PAGE
            ;1972   .TOC "    ERLOOP ROUTINE"
U 115B, 0818,0038,0980,0800,0000,105E   ;1973 ERLOOP: D_K[.2],J/CALLCON   ; ERRLOOP CALL
            ;1974
```

```
            :1975   .PAGE
            :1976   .TOC "    ERROR1 ROUTINE"
U 115C. 0810.0038.0180.0978.0000.115D  :1977 ERROR1: D_RC[0F]
U 115D. 0819.0034.4D80.0800.0000.104E  :1978  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  :1979 104E: D_D.OR.K[.4],J/CALLCON
            :1980
```

```
            ;1981  .PAGE
            ;1982  .TOC "    ERROR1 WITH PARAMETER"
            ;1983 ;**
            ;1984 ; FUNCTIONAL DESCRIPTION:
            ;1985 ;
            ;1986 ; This subroutine takes  as parameter the number of arguments
            ;1987 ; to be printed to the console in the case of an error logout.
            ;1988 ;
            ;1989 ; CALLING CONSTRAINT:
            ;1990 ;
            ;1991 ; =0
            ;1992 ;INPUTS:
            ;1993 ;
            ;1994 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;1995 ;--
            ;1996 =0
            ;1997 ERR1.ARG:
U 103E, 0000,003D,0180,0800,0000,1164  ;1998  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 103F, 0000,003C,0180,0800,0000,115C  ;1999  J/ERROR1  ; Go to ERROR1
            ;2000 ;
            ;2001 ;
            ;2002
```

```
            ;2003  .PAGE
            ;2004  .TOC "    ERROR2 ROUTINE"
U 115E. 0810,0038,0180,0978,0000,115F  ;2005 ERROR2: D_RC[0F]
U 115F. 0819,0034,4D80,0800,0000,105A  ;2006  D_D.AND.K[.FF00]
U 105A. 0819,0030,D580,0800,0000,105E  ;2007 105A: D_D.OR.K[.6],J/CALLCON
            ;2008
```

```
            ;2009  .PAGE
            ;2010  .TOC "   ERROR 2 WITH PARAMETER"
            ;2011 ;++
            ;2012 ; FUNCTIONAL DESCRIPTION:
            ;2013 ;
            ;2014 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2015 ; except that in this case after setting the number of arguments
            ;2016 ; too the console, control is transferred to routine ERROR2.
            ;2017 ;
            ;2018 ; INPUTS:
            ;2019 ;
            ;2020 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2021 ;
            ;2022 ;--
            ;2023 =0
            ;2024 ERR2.ARG:
U 1040, 0000,003D,0180,0800,0000,1164  ;2025  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1041, 0000,003C,0180,0800,0000,115E  ;2026  J/ERROR2  ; Go to ERROR2
            ;2027 ;
            ;2028 ;
            ;2029
```

```
            ;2030  .PAGE
            ;2031  .TOC "   CALLCON ROUTINE"
            ;2032 105E:
            ;2033 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2034  ID[TXDB]_D.J/CALLCON  ; CALL THE CONSOLE
            ;2035
```

```
            ;2036  .PAGE
            ;2037  .TOC "   SCRATCH PAD LOCATIONS"
            ;2038 ;*
            ;2039 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2040 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2041 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;2042 13F1: NOP
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2043 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2044 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2045 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2046 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2047 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2048 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2049 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2050 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2051 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2052 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2053 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2054 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2055 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;2056 13FF: NOP
U 1155, 0000.003C.0180.0800.0000.12AA  ;2057 1155: NOP
U 12AA, 0000.003C.0180.0800.0000.1160  ;2058 12AA: NOP
            ;2059
```

```
            ;2060   .PAGE
            ;2061   .TOC "    SUBTEST ROUTINE"
            ;2062 ;+
            ;2063 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2064 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2065 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2066 ; TYPING IT.
            ;2067 ;-
U 1160, 0810,0038,4D80,0978,0000,1161   ;2068 SUBTST: D_RC[0F],KMX/.FF00
U 1161, 0019,4016,4D80,09F8,0000,0001   ;2069   RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2070
```

```
            ;2071  .PAGE
            ;2072  .TOC "   SUBTEST 1 ROUTINE"
            ;2073 ;*****************************************************************
            ;2074 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2075 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2076 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2077 ;-
            ;2078 SUBTEST1:
U 1162, 0810,0038,0180,0978,0000,1163  ;2079  D_RC[0F]
U 1163, 0019,0032,4D80,09F8,0000,0001  ;2080  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2081
```

```
            ;2082  .PAGE
            ;2083  .TOC "    SETRCF ROUTINE"
            ;2084 ;*
            ;2085 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2086 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2087 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2088 ;-
U 1164, 0010,0038,01C0,0978,0000,1165  ;2089 SETRCF: Q_RC[0F]
U 1165, 0019,2034,4DC0,0800,0000,1166  ;2090  Q_Q.AND.K[.FF00]
U 1166, 0019,2032,1D80,09F8,0000,0001  ;2091  RC[0F]_Q.OR.SC,RETURN1
            ;2092
```

```
            ;2093  .PAGE
            ;2094  .TOC "   ENBLFLT ROUTINE"
            ;2095 ;*****************************************************************
            ;2096 ;*
            ;2097 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2098 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2099 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2100 ;
            ;2101 ; D AND SC GET CLOBBERED
            ;2102 ;-
            ;2103 ;*****************************************************************
U 1167, 0F00,003C,2180,0800,0084,7168  ;2104 ENBFLT: SC_K[.14],D_0
U 1168, 0000,003C,0580,0800,0084,B169  ;2105  SC_SC-K[.1]
U 1169, 0801,001C,0180,0800,0000,116A  ;2106  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 116A, 0600,003C,6D80,3C00,0000,116B  ;2107  ID[FAULT]_D,D_D.RIGHT
U 116B, 0000,003E,6D80,3C00,0000,0001  ;2108  ID[FAULT]_D,RETURN1
            ;2109
```

```
          ;2110  .PAGE
          ;2111  .TOC "    NOINTR ROUTINE"
          ;2112 ;*****************************************************************
          ;2113 ;+
          ;2114 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
          ;2115 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
          ;2116 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
          ;2117 ; IF ANY INTERRUPTS OCCURRED:
          ;2118 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
          ;2119 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
          ;2120 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
          ;2121 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
          ;2122 ;  RETURN1 IS USED TO GET BACK
          ;2123 ; ELSE
          ;2124 ;  RETURN2 IS USED
          ;2125 ;
          ;2126 ; THE D AND Q REGISTERS GET CLOBBERED.
          ;2127 ;
          ;2128 ;-
          ;2129 ;*****************************************************************
          ;2130 =0
U 1042, 0000.003D.0180.0800.0000.1171  ;2131 NOINTR: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1043, 0000.003C.35F0.2C00.0000.116C  ;2132  Q_ID[VECTOR]
U 116C, 0019.2034.8180.0800.0010.116D  ;2133  ALU_Q.AND.K[.3FF],CLK.UBCC
U 116D, 0000.013C.0180.0800.0000.1044  ;2134  Z?
U 1044, 0000.003C.0180.0800.0000.1175  ;2135 =0 J/ERRFLT   ; INTR. OCCURRED,SET FLAG,GET DATA
U 1045, 0000.003E.0180.0800.0000.0002  ;2136  RETURN2    ; NO INTR, RETURN2
          ;2137 =0
U 1046, 0000.003D.0180.0800.0000.1171  ;2138 WRTOUT: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1047, 0000.003C.35F0.2C00.0000.116E  ;2139  Q_ID[VECTOR]
U 116E, 0819.2034.8180.0800.0000.116F  ;2140  ALU_Q.AND.K[.3FF],D_ALU
U 116F, 0019.0020.A580.0800.0010.1170  ;2141  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 1170, 0000.013C.0180.0800.0000.1048  ;2142  Z?
U 1048, 0000.003C.0180.0800.0000.1175  ;2143 =0 J/ERRFLT   ; WRONG VECTOR,SET FLAG,GET DATA
U 1049, 0000.003E.0180.0800.0000.0002  ;2144  RETURN2    ; OK, RETURN2
U 1171, 0F00.003C.0180.0800.0000.1172  ;2145 STROBE: D_0     ; SET LOWEST IPL ACTIVE
U 1172, 0000.003C.3D80.3C00.0000.1173  ;2146  ID[PSL]_D    ;
U 1173, 0000.003C.0180.0800.4000.1174  ;2147  IEK/ISTR    ; STROBE INTERRUPT
U 1174, 0000.003E.0180.0800.0000.0001  ;2148  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 1175, 0018.0038.0580.09A8.0000.1176  ;2149 ERRFLT: RC[5]_K[.1]   ; Set Flag
U 1176, 0001.203C.65F0.2DC0.0000.1177  ;2150  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 1177, 0001.203C.6DF0.2DB8.0000.1178  ;2151  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 1178, 0001.203E.0180.09B0.0000.0001  ;2152  RC[6]_Q,RETURN1    ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
          ;2153
          ;2154
```

```
            ;2155  .PAGE
            ;2156  .TOC "   GENREATE CONSTANT ROUTINE (DCE)"
            ;2157 ;////////////////////////////////////////////////////////////////
            ;2158 ;*
            ;2159 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2160 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2161 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2162 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2163 ;
            ;2164 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2165 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2166 ;
            ;2167 ;   =0 D_0,CALL,J/DC5
            ;2168 ;    NOP
            ;2169 ;   =0 CALL,J/DC5
            ;2170 ;-
            ;2171
U 1179.  0018,0038,C180,0800,0000,117C  ;2172 DCE: ALU_K[.FFFF],J/S10
            ;2173
U 117A.  0118,0038,1B80,0800,0000,117E  ;2174 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 117B.  0118,0038,0B80,0800,0000,117E  ;2175 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 117C.  0118,0038,0780,0800,0000,117E  ;2176 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 117D.  0118,0038,C380,0800,0000,117E  ;2177 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
U 117E.  0100,003E,0380,0800,0000,0001  ;2178 SX: D_D.LEFT2,SI/7,RETURN1 ; EXIT
            ;2179
            ;2180
```

```
            ;2181   .PAGE
            ;2182   .TOC "    DELAY ROUTINE"
U 117F, 0818,0038,0180,0800,0000,105E  ;2183 DELAY: D_K[.8],J/CALLCON
            ;2184
```

```
:2185  .PAGE
:2186  .TOC " DIAGNOSTIC SPECIFIC ROUTINES"
```

```
            ;2187  .PAGE
            ;2188  .TOC "   CHECK UTRAP"
            ;2189 ;++
            ;2190 ;FUNCTIONAL DESCRIPTION:
            ;2191 ;
            ;2192 ; This subroutine is used to check wether a Utrap occured.
            ;2193 ; It takes the contents of the Save Utrap flags register
            ;2194 ; R[0E], and compares them to the contrnts of the Utrap
            ;2195 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2196 ; according to the results of the test  (i.e., contents of
            ;2197 ; these registers are equal or not).
            ;2198 ; CALLING CONSTRAINT:
            ;2199 ;
            ;2200 ; =00
            ;2201 ;
            ;2202 ; INPUTS:
            ;2203 ;
            ;2204 ; R[0E]- Saved Utrap Mask
            ;2205 ; R[0F]- Expected Utrap Mask
            ;2206 ;
            ;2207 ;--
            ;2208 CHECK_UTRAP:
U 1180, 0800,003C,0180,0A70,0000,1181  ;2209  D_R[0E]   ; Reg D is loaded with the trap mask
U 1181, 0001,003C,0180,09C0,0000,1182  ;2210  RC[08]_D  ; Save expected Utrap flag for error logout
U 1182, 0800,003C,0180,0A78,0000,1183  ;2211  D_R[0F]   ; Get recieved Utrap flag to D reg
U 1183, 0001,003C,0180,09B8,0000,1184  ;2212  RC[07]_D  ; Save in RC[07]
            ;2213  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1184, 000D,0020,0180,0A70,0010,1185  ;2214        CLK.UBCC
U 1185, 0003,013C,0180,0AF8,0000,104A  ;2215  Z?,R[0F]_0  ; Branch if no traps
U 104A, 0000,003E,0180,0800,0000,0002  ;2216 =0 RETURN2   ; Utrap occured
U 104B, 0000,003E,0180,0800,0000,0001  ;2217  RETURN1    ; No Utrap  return
            ;2218
```

```
            ;2219  .PAGE
            ;2220  .TOC "    Set Trap Mask"
            ;2221 ;**
            ;2222 ; FUNCTIONAL DESCRIPTION:
            ;2223 ;
            ;2224 ; This routine is used to set a bit in the Micro Trap Mask
            ;2225 ; R[0F].
            ;2226 ;
            ;2227 ; CALLING CONSTRAINT:
            ;2228 ;
            ;2229 ; =0
            ;2230 ;
            ;2231 ; INPUTS:
            ;2232 ;
            ;2233 ; SC - Contains bit number to set.
            ;2234 ;
            ;2235 ; OUTPUTS:
            ;2236 ;
            ;2237 ; rc[0E] - Contains the current trap mask
            ;2238 ;
            ;2239 ; SIDE EFFECTS:
            ;2240 ;
            ;2241 ; d regsiter is destroyed
            ;2242 ;--
            ;2243 SET_TRAP_MASK:
U 1186, 0800,003C,0180,0A78,0000,1187  ;2244  d_r[0f]
U 1187, 0801,001C,0180,0AF8,0000,1188  ;2245  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1188, 0001,003E,0180,0AF0,0000,0001  ;2246  r[0E]_d,return1    ; Save mask and return
            ;2247
```

```
            ;2248  .PAGE
            ;2249  .TOC "   Disable Cache Routine"
            ;2250 ;**
            ;2251 ; FUNCTIONAL DESCRIPTION:
            ;2252 ;
            ;2253 ; This routine disables cache hits by setting bits <16:15>
            ;2254 ; in the maintenance register.
            ;2255 ;
            ;2256 ; CALLING SEQUENCE:
            ;2257 ;
            ;2258 ; =0
            ;2259 ;
            ;2260 ; SIDE EFFECTS:
            ;2261 ;
            ;2262 ; D & Q Registers destroyed
            ;2263 ;--
            ;2264 DISABLE.CACHE:
U 1189, 0818,0038,4580,0800,0000,118A  ;2265  d_k[.8000]  ; ... and seed constant
U 118A, 0500,003C,0180,0800,0000,118B  ;2266  d_d.left  ; Generate final constant
U 118B, 0819,0030,4580,0800,0000,118C  ;2267  d_d.or.k[.8000]  ; ... = 00018000
U 118C, 0000,003E,7580,3C00,0000,0001  ;2268  id[maint]_d,return1 ; Disable cache and return
            ;2269
```

```
            ;2270  .PAGE
            ;2271  .TOC "   Initialize the SBI"
            ;2272 ;++
            ;2273 ; FUNCTIONAL DESCRIPTION:
            ;2274 ;
            ;2275 ; This routine performs the following functions:
            ;2276 ;
            ;2277 ;  Executes an SBI Unjam
            ;2278 ;  Clears the SBI Fault Latch
            ;2279 ;  Clears the SBI Error Register
            ;2280 ;
            ;2281 ; CALLING CONSTRAINT:
            ;2282 ;
            ;2283 ; =0
            ;2284 ;
            ;2285 ; SIDE EFFECTS:
            ;2286 ;
            ;2287 ; D & Q Register destroyed
            ;2288 ;--
            ;2289 =0
            ;2290 SBI.INIT:
U 104C, 0000,003D,0180,0800,0000,11DB  ;2291  call[sbi.unjam]  ; Unjam the SBI
U 104D, 0818,0038,C180,0800,0000,118D  ;2292  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 118D, 0801,0028,6580,3C00,0000,118E  ;2293  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 118E, 0F00,003C,6D80,3C00,0000,118F  ;2294  id[fault]_d,d_0
U 118F, 0000,003C,6D80,3C00,0000,1190  ;2295  id[fault]_d
U 1190, 0000,003E,6580,3C00,0000,0001  ;2296  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2297
            ;2298
```

```
            ;2299  .PAGE
            ;2300  .TOC "   FIND MS780-E MEMORY"
            ;2301 ;**
            ;2302 ; FUNCTIONAL DESCRIPTION:
            ;2303 ;
            ;2304 ;          The diagnostic is designed to handle up to 4 (four)
            ;2305 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2306 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2307 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2308 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2309 ;     MS780-E/H's found minus one. Also, it is to be noted that the
            ;2310 ;     first MS780-E found will have its starting address set to 0
            ;2311 ;     (zero).
            ;2312 ;     All the other MS780-E/H's found will have their starting address
            ;2313 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2314 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2315 ;     apropriate action.   Register RC[0E] is used as a pointer to the
            ;2316 ;     current MS780-E unit under test.
            ;2317 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
            ;2318 ;     set to maximum.
            ;2319 ;
            ;2320 ; CALLING CONSTRAINT:
            ;2321 ;
            ;2322 ; =00
            ;2323 ;
            ;2324 ; OUTPUTS:
            ;2325 ;
            ;2326 ; rc[0e] - Contains address of Configuration Register
            ;2327 ; r[0]  - Contains TR level
            ;2328 ;
            ;2329 ; SIDE EFFECTS:
            ;2330 ;
            ;2331 ; D register is destroyed.
            ;2332 ;--
            ;2333 =0
            ;2334 FIND.MS780E:
U 1050, 0000,003D,0180,0800,0000,104C  ;2335  call[sbi.init]  ; Make sure SBI is enabled
U 1051, 0003,003C,0180,0988,0000,1191  ;2336  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1191, 0818,0038,1980,0800,0000,1192  ;2337  d_k[zero]  ; Clear MS780-E/H counter
U 1192, 0000,003C,F980,3C00,0000,1193  ;2338  id[3e]_d  ; ...
U 1193, 0018,0038,0580,0A80,0000,1194  ;2339  r[0]_k[.1]  ; Init TR counter
U 1194, 0F03,003C,0180,09F0,0000,1052  ;2340  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2341      ; ...CNFG A Reg
            ;2342 =0
            ;2343 FIND.MEM.LOOP:
U 1052, 0800,003D,0180,0A00,0000,11BA  ;2344  d_r[0],call[generate.io]; Generate address of TR level
U 1053, 0001,003C,0180,09F0,0200,1054  ;2345  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1054, 0000,003D,8980,0800,0084,7186  ;2346 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2347  d[long]_cache.p, ; Read the specified tr level
U 1055, 0000,003C,0180,C970,0000,1195  ;2348   ic_rc[0e]  ; ...
U 1195, 0000,003C,0180,0A78,0010,1196  ;2349  alu_r[0f],clk.ubcc ; Check for no response
U 1196, 0001,013C,0180,0880,0082,1056  ;2350  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 1056, 0000,003C,0180,0800,0000,1197  ;2351 =0 j/check.adpt.code ; Check for a memory
U 1057, 0000,003C,0180,0800,0000,11B6  ;2352  j/find.mem.lp1  ; Try next tr
            ;2353 CHECK.ADPT.CODE:
```

```
U 1197, 0000,003C,1D80,0800,1404,7198  ;2354  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1198, 0000,163C,0180,0800,0000,1008  ;2355  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,7199  ;2356 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,719A  ;2357  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11B6  ;2358  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11B6  ;2359  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,719F  ;2360  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11B6  ;2361  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,11A3  ;2362  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,11A3  ;2363  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2364   ;
            ;2365   ; Found an MS780-A or -C. Set the starting address
            ;2366   ; to maximum.
            ;2367   ;
U 1199, 0818,0038,5D80,0800,0000,119B  ;2368 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 119A, 0818,0038,0580,0800,0000,119B  ;2369 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2370 MS780.COMMON:
U 119B, 0819,0030,7180,0800,0000,119C  ;2371  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 119C, 0000,003C,01F8,0803,0080,D19D  ;2372  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 119D, 0D00,003C,0180,0800,0000,119E  ;2373  d_dal.sc  ; Put data in bits<29:14>
            ;2374  cache.p_d[long], ; Set the starting address to maximum
U 119E, 0000,003C,0180,A800,0000,11B6  ;2375       j/find.mem.lp1 ; and continue TR scan
            ;2376   ;
            ;2377   ; Found an MA780. Set the starting address to maximum
            ;2378   ;
            ;2379 MA780.MAX:
U 119F, 0818,0038,39F8,0803,0000,11A0  ;2380  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11A0, 0D00,003C,0180,0803,0000,11A1  ;2381  d_dal.sc,va_va+4 ; Put in proper position
U 11A1, 0000,003C,0180,0803,0000,11A2  ;2382  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2383  cache.p_d[long], ; Set starting address to maximum
U 11A2, 0000,003C,0180,A800,0000,11B6  ;2384       j/find.mem.lp1
            ;2385 ;
            ;2386 ; Found an MS780E
            ;2387 ;
            ;2388 FOUND.MS780E:
U 11A3, 0000,003C,F9F0,2C00,0000,11A4  ;2389  q_id[3e]  ; Set up to see how many MS780-E's
            ;2390  alu_q.xor.k[.3], ; Are there Maximum units?
U 11A4, 0019,2020,0D80,0800,0010,11A5  ;2391   clk.ubcc ; Check condition codes
U 11A5, 0000,013C,0180,0800,0000,1058  ;2392  z?   ; Are we at maximum units for system
U 1058, 0000,003C,0180,0800,0000,11A6  ;2393 =0 J/ms780e.tst  ; No go find how many units ther are
U 1059, 0818,0038,C180,0800,0000,105C  ;2394  d_k[.ffff]  ; Yes set address to maximum
U 105C, 0000,003D,0180,0800,0000,11BE  ;2395 =0 call[set.start.addr] ; .....
U 105D, 0000,003C,0180,0800,0000,11B6  ;2396  j/find.mem.lp1  ; Go check to see if we are done
            ;2397 ;
            ;2398 MS780E.TST:
            ;2399  alu_rc[01],  ; Check "Found MS780E Flag"
U 11A6, 0010,0038,0180,0908,0010,11A7  ;2400   clk.ubcc ; Check condition codes
U 11A7, 0810,0138,0180,0970,0000,1060  ;2401  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2402 =0 j/not.first.ms780e, ; No
U 1060, 0818,0038,C180,0800,0000,1064  ;2403   d_k[.ffff] ; Set starting address
U 1061, 0001,003C,0180,0988,0000,11A8  ;2404  rc[01]_d  ; Yes put base address in rc[01]
U 11A8, 0818,0038,7980,0800,0000,11A9  ;2405  d_k[.30]  ; Set up SC to point to first unit
U 11A9, 0019,0014,F580,0800,0082,11AA  ;2406  alu_d+k[.a],sc_alu ; ...
U 11AA, 0810,0038,0180,0908,0000,11AB  ;2407  d_rc[01]  ; Put base address of unit in D
U 11AB, 0000,003C,0180,3400,0000,11AC  ;2408  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11AC, 0F00,003C,0180,0800,0000,1062  ;2409  d_0   ; Prepare to set starting address to Zero
U 1062, 0000,003D,0180,0800,0000,11BE  ;2410 =0 call[set.start.addr] ; Go do it
        ;2411  j/find.mem.lp1,  ; Check to see if we are done
U 1063, 0003,003C,0180,09E8,0000,11B6  ;2412   rc[0d]_0 ; ....
        ;2413 =0
        ;2414 NOT.FIRST.MS780E:
U 1064, 0000,003D,0180,0800,0000,11BE  ;2415  call[set.start.addr] ; Go set starting address to maximum
U 1065, 0000,003C,F9F0,2C00,0000,11AD  ;2416  q_id[3e]  ; Get the number of MS780-Es found
U 11AD, 0819,2014,0580,0800,0000,11AE  ;2417  d_q+k[.1]  ; Increment this counter
U 11AE, 0000,003C,F980,3C00,0000,11AF  ;2418  id[3e]_d  ; Save the counter
U 11AF, 0018,0038,11C0,0800,0000,11B0  ;2419  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2420     ; ...were the I/O base addresses will be stored
U 11B0, 081D,2000,7980,0800,0000,11B1  ;2421  d_q-d,k[.30]  ; ... adjust pointer
U 11B1, 0819,0014,7980,0800,0000,11B2  ;2422  d_d+k[.30]  ; Point the SC to the ID register
U 11B2, 0000,003C,F580,0800,0000,11B3  ;2423  k[.a]   ; ...to receive I/O base address
U 11B3, 0019,0014,F580,0800,0082,11B4  ;2424  alu_d+k[.a],sc_alu ; ...
U 11B4, 0810,0038,0180,0970,0000,11B5  ;2425  d_rc[0e]  ; Get base address to d
U 11B5, 0000,003C,0180,3400,0000,11B6  ;2426  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2427   ;
        ;2428   ; Try next tr
        ;2429   ;
        ;2430 FIND.MEM.LP1:
U 11B6, 0818,0014,0580,0A80,0000,11B7  ;2431  r[0]_la+k[.1],d_alu ; Increment TR level
        ;2432  alu_d.and.k[.f],
U 11B7, 0019,0034,6180,0800,0010,11B8  ;2433      clk.ubcc  ; Check if all done
U 11B8, 0000,013C,0180,0800,0000,1066  ;2434  z?   ; Branch if yes
U 1066, 0000,003C,0180,0800,0000,1052  ;2435 =0 j/find.mem.loop  ; Try next TR
U 1067, 0810,0038,0180,0908,0010,11B9  ;2436  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11B9, 0000,013C,0180,0800,0000,1068  ;2437  z?   ; Check condition codes
U 1068, 0001,003E,0180,09F0,0000,0002  ;2438 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1069, 0000,003E,0180,0800,0000,0001  ;2439  return1   ; No MS780E found
        ;2440
```

```
            ;2441  .PAGE
            ;2442  .TOC "   Generate IO Address"
            ;2443 ;++
            ;2444 ; FUNCTIONAL DESCRIPTION:
            ;2445 ;
            ;2446 ; This routine is used to generate an SBI Configuration Register
            ;2447 ; address from a TR level.
            ;2448 ;
            ;2449 ; CALLING CONSTRAINT:
            ;2450 ;
            ;2451 ; =0
            ;2452 ;
            ;2453 ; INPUTS:
            ;2454 ;
            ;2455 ; D - Contains TR level
            ;2456 ;
            ;2457 ; OUTPUTS:
            ;2458 ;
            ;2459 ; D - Contains SBI IO address
            ;2460 ;--
            ;2461 GENERATE.IO:
U 11BA, 0000,003C,89F8,0800,0084,71BB  ;2462  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11BB, 0D00,003C,8D80,0800,0084,71BC  ;2463  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11BC, 0000,003C,0980,0800,0084,B1BD  ;2464  sc_sc-k[.2]  ; insert bit 29
U 11BD, 0801,001E,0180,0800,0000,0001  ;2465  d_d.ornot.mask,return1 ; and return
            ;2466
            ;2467
```

```
            ;2468  .PAGE
            ;2469  .TOC "    Set Starting Address"
            ;2470 ;++
            ;2471 ; FUNCTIONAL DESCRIPTION:
            ;2472 ;
            ;2473 ; This routine is used to set the starting address of the
            ;2474 ; memory to the specified value.
            ;2475 ;
            ;2476 ; CALLING CONSTRAINT:
            ;2477 ;
            ;2478 ; =0
            ;2479 ;
            ;2480 ; INPUTS:
            ;2481 ;
            ;2482 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2483 ;     (B Register Image divided by 2**16)
            ;2484 ; rc[0] - Contains Address of Register A
            ;2485 ;
            ;2486 ; OUTPUTS:
            ;2487 ;
            ;2488 ; d - Contains aligned starting address (B Register Image)
            ;2489 ;
            ;2490 ; SIDE EFFECTS:
            ;2491 ;
            ;2492 ; d,q,va, and sc are destroyed
            ;2493 ;--
            ;2494 SET.START.ADDR:
U 11BE, 0010,0038,6580,0970,0284,71BF  ;2495  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11BF, 0000,003C,01F8,0803,0000,11C0  ;2496  va_va+4,q_0  ; Set address to Register B
            ;2497  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11C0, 0D00,003C,0980,0800,0084,B1C1  ;2498      sc_sc-k[.2]  ; Setup to insert bit 14
U 11C1, 0801,001C,0180,0800,0000,11C2  ;2499  d_d.ornot.mask  ; Insert Write Enable bit
U 11C2, 0000,003E,0180,A800,0000,0001  ;2500  cache.p_d[long],return1 ; Set the starting address and return
            ;2501
```

```
            ;2502  .PAGE
            ;2503  .TOC "   ENABLE NEXT MS780-E"
            ;2504 ;++
            ;2505 ; FUNCTIONAL DESCRIPTION:
            ;2506 ;
            ;2507 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2508 ;    They will be tested  separately, according to the TR level at which
            ;2509 ;    they are located,  starting with the one  at the lowest level first.
            ;2510 ;          When  a test has finished  with the first unit, it will have to call
            ;2511 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2512 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2513 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2514 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2515 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2516 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2517 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2518 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2519 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2520 ;
            ;2521 ; CALLING CONSTRAINT:
            ;2522 ;
            ;2523 ;  =00
            ;2524 ;
            ;2525 ;--
            ;2526 ENABLE.NEXT.MS780E:
U 11C3, 0000,003C,F9F0,2C00,0000,11C4  ;2527  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2528  ALU_Q,    ; Check to see if it is zero
U 11C4, 0001,203C,0180,0800,0010,11C5  ;2529     CLK.UBCC  ; Check Condition Codes
U 11C5, 0000,013C,01B0,0800,0000,106A  ;2530  Z?    ; ...for zero
U 106A, 0000,003C,0180,0800,0000,11C6  ;2531 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 106B, 0000,003E,0180,0800,0000,0002  ;2532  RETURN2   ; Zero No more units to test
            ;2533 ENABLE.NEXT:
U 11C6, 0818,0038,C180,0800,0000,106C  ;2534  D_K[.FFFF]  ; Load D with Maximum Starting address
U 106C, 0000,003D,0180,0800,0000,11BE  ;2535 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 106D, 0818,0038,7980,0800,0000,11C7  ;2536  D_K[.30]  ; Prepare to point to the ID register holding
            ;2537     ; ...the I/O Base address of the next MS780-E
U 11C7, 0819,0014,1180,0800,0000,11C8  ;2538  D_D+K[.4]  ; ...
U 11C8, 0000,003C,F580,0800,0000,11C9  ;2539  K[.A]    ; ...
U 11C9, 0819,0014,F580,0800,0000,11CA  ;2540  D_D+K[.A]  ; ...
U 11CA, 0000,003C,F9F0,2C00,0000,11CB  ;2541  Q_ID[3E]  ; Get MS780-E Counter
            ;2542  D_D-Q,    ; D now holds the pointer to the ID register
U 11CB, 081D,0000,0180,0800,0082,11CC  ;2543    SC_ALU  ; ...
U 11CC, 0000,003C,01F0,2400,0000,11CD  ;2544  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11CD, 0001,203C,0180,09F0,0000,11CE  ;2545  RC[0E]_Q  ; Save it also in RC[0E]
U 11CE, 0F03,003C,0180,09E8,0000,106E  ;2546  RC[0D]_0,D_0  ; Set starting address to zero
U 106E, 0000,003D,0180,0800,0000,11BE  ;2547 =0 CALL[SET.START.ADDR] ; ....
U 106F, 0F00,003C,F9F0,2C00,0000,11CF  ;2548  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11CF, 081D,2008,0180,0800,0000,11D0  ;2549  D_Q-D-1   ; Subtract 1 from total
U 11D0, 0000,003C,F980,3C00,0000,1070  ;2550  ID[3E]_D  ; Adjust the pointer
U 1070, 0000,003D,0180,0800,0000,1218  ;2551 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 1071, 0000,003E,0180,0800,0000,0001  ;2552  RETURN1   ; Tell caller another unit was found
```

```
            ;2553  .PAGE
            ;2554  .TOC "   Select Controller"
            ;2555 ;++
            ;2556 ; FUNCTIONAL DESCRIPTION:
            ;2557 ;
            ;2558 ; This routine is used to put the memory system into the
            ;2559 ; specified configuration with respect to controller select.
            ;2560 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2561 ; a RETURN2 is made.
            ;2562 ;
            ;2563 ; CALLING CONSTRAINT:
            ;2564 ;
            ;2565 ; =00
            ;2566 ;
            ;2567 ; INPUTS:
            ;2568 ;
            ;2569 ; d - Contains value to write to bits<2:0> of Register A
            ;2570 ; rc[0e] - Address of Register A
            ;2571 ;
            ;2572 ; SIDE EFFECTS:
            ;2573 ;
            ;2574 ; sc,d,va are destroyed
            ;2575 ;--
            ;2576 SELECT.CNTRLR:
U 11D1, 0010.003B.0180.0970.0284.71D2  ;2577  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11D2, 0801.001C.0180.0800.0000.11D3  ;2578  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11D3, 0000.003C.0180.A800.0000.11D4  ;2579  cache.p_d[long]  ; Write the configuration Register
U 11D4, 0818.003B.5D80.0800.0000.11D5  ;2580  d_k[.7]   ; Get Misconfiguration bits
U 11D5, 0000.003C.61F8.0800.0084.71D6  ;2581  sc_k[.F],q_0  ; ...
U 11D6, 0D00.003C.0180.0800.0000.11D7  ;2582  d_dal.sc  ; ... D = x38000
U 11D7, 0000.003C.01E0.0800.0000.11D8  ;2583  q_d   ; Save mask
U 11D8, 0000.003C.0180.C800.0000.11D9  ;2584  d[long]_cache.p  ; Read CNFG A Reg and
            ;2585  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11D9, 001D.2034.0180.0800.0010.11DA  ;2586   clk.ubcc ; ...
U 11DA, 0000.013C.0180.0800.0000.1072  ;2587  z?   ; Branch if no
U 1072, 0000.003E.0180.0800.0000.0001  ;2588 =0 RETURN1   ; Bad configuration
U 1073, 0000.003E.0180.0800.0000.0002  ;2589  RETURN2   ; Configuration ok
            ;2590
```

```
;2591  .PAGE
;2592  .TOC "   SELECT LOWER CONTROLLER"
;2593 ;*************************************************************
;2594 ;++
;2595 ;
;2596 ; Functional Description:
;2597 ; ----------------------
;2598 ;
;2599 ; This subroutine can be called to select the lower
;2600 ; Controller on a MS780-E/H.
;2601 ; If the Lower controller is misconfigured then a
;2602 ; RETURN1 will be made.  Otherwise a RETURN2
;2603 ;
;2604 ; Calling Constraint: =00
;2605 ; ------------------
;2606 ;
;2607 ;
;2608 ; Inputs:
;2609 ; ------
;2610 ;
;2611 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2612 ;
;2613 ; Outputs:
;2614 ; -------
;2615 ;
;2616 ; RC[0D] will have the address of CNFG Register C
;2617 ;       ( 2000x008 )
;2618 ;
;2619 ; Side Effects:
;2620 ; ------------
;2621 ;
;2622 ; Contents of the following registers will lost:
;2623 ;
;2624 ; D
;2625 ;
;2626 ; The Lower controller on the MS780-E/H will be configured
;2627 ; when the subroutine is exitted with a RETURN2
;2628 ;--
;2629 ;*************************************************************
;2630 =00
;2631 SELECT.LOWER:
;2632  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,11D1  ;2633   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2634  RETURN1    ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2635  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2636  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2637   RETURN2   ; Let caller know that the controller
;2638       ; ...was configured properly
```

```
;2639  .PAGE
;2640  .TOC "   SELECT UPPER CONTROLLER"
;2641 ;*************************************************************
;2642 ;**
;2643 ;
;2644 ; Functional Description:
;2645 ; ----------------------
;2646 ;
;2647 ; This subroutine can be called to select the upper
;2648 ; Controller on a MS780-E/H.
;2649 ; If the Upper controller is misconfigured then a
;2650 ; RETURN1 will be made.  Otherwise a RETURN2
;2651 ;
;2652 ; Calling Constraint: =00
;2653 ; ------------------
;2654 ;
;2655 ;
;2656 ; Inputs:
;2657 ; ------
;2658 ;
;2659 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2660 ;
;2661 ; Outputs:
;2662 ; -------
;2663 ;
;2664 ; RC[0D] will have the address of CNFG Register D
;2665 ;        ( 2000x010 )
;2666 ;
;2667 ; Side Effects:
;2668 ; ------------
;2669 ;
;2670 ; Contents of the following registers will lost:
;2671 ;
;2672 ; D
;2673 ;
;2674 ; The Upper controller on the MS780-E/H will be configured
;2675 ; when the subroutine is exitted with a RETURN2
;2676 ;--
;2677 ;*************************************************************
;2678 =00
;2679 SELECT.UPPER:
;2680  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,11D1  ;2681   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2682  RETURN1    ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2683  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2684  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2685   RETURN2   ; Let caller know that the controller
;2686       ; ...was configured properly
;2687
```

```
            ;2688  .PAGE
            ;2689  .TOC "   CLRSBI ROUTINE"
            ;2690 CLRSBI:
            ;2691 ;================================================================================
            ;2692 ;////////////////////////////////////////////////////////////////////////////////
            ;2693 ;*
            ;2694 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2695 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2696 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2697 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2698 ;-
            ;2699 SBI.UNJAM:
U 11DB. 0818,0038,6580,8000,0010,1074  ;2700 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2701 =0
U 1074. 0819,0100,0580,8000,0010,1074  ;2702 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 1075. 0818,0038,6580,8800,0010,1076  ;2703  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2704 =0
U 1076. 0819,0100,0580,8800,0010,1076  ;2705 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 1077. 0818,0038,6580,8000,0010,1078  ;2706  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2707 =0
U 1078. 0819,0100,0580,8000,0010,1078  ;2708 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 1079. 0000,003E,0180,0800,0000,0001  ;2709  RETURN1    ; DONE.
            ;2710
```

```
            ;2711  .PAGE
            ;2712  .TOC "   CPUCA ROUTINE"
            ;2713 ;*****************************************************************
            ;2714 ;*
            ;2715 ;       THIS ROUTINE PLACES THE EXPECTED DATA PATTERN FOR
            ;2716 ; CPU COMMAND ADDRESS AS IT APPEARS IN THE SILO.
            ;2717 ; IT GENERATES THE ID AND TAG FIELDS ONLY, I.E. IT IS
            ;2718 ; UP TO THE CALLING ROUTINE TO "OR IN" THE FUNCTION CODE.
            ;2719 ;
            ;2720 ; D IS RETURNED WITH THE VALUE 00XX,XXX0,1100,XXXX,XXXX,XXXX,XXXX
            ;2721 ;  X'S ARE DEPENDENT ON CPU'S ID AND TR LEVEL
            ;2722 ; SC , Q,RC[0E]-RC[0] GET CLOBBERED
            ;2723 ;-
            ;2724 ;*****************************************************************
            ;2725 =0
U 107A, 0000,003C,8980,0800,0084,707B  ;2726 CPUCA: SC_K[.D]
U 107B, 0003,001C,0180,0AF8,0000,107C  ;2727  R[0F]_NOT.MASK
U 107C, 0000,003D,0180,0800,0000,11DF  ;2728 =0 CALL,J/SILCLR
U 107D, 0000,003C,0180,0800,0000,107E  ;2729  NOP
            ;2730 =0 D_0,
U 107E, 0F00,003D,0180,0800,0000,11BA  ;2731   CALL[GENERATE.IO]
U 107F, 0001,003C,7180,3C00,0200,1080  ;2732  ID[COMP]_D.VA_D
U 1080, 0000,003D,0180,C800,0000,1086  ;2733 =0 CALL,J/SLOLCK,MCT/READ.P
U 1081, 0000,003C,0180,0800,0000,1082  ;2734  NOP
U 1082, 0000,003D,0180,0800,0000,11EB  ;2735 =0 CALL,J/CLCPTO
U 1083, 0810,0038,0180,0958,0000,11DC  ;2736  D_RC[0B]
U 11DC, 0000,003C,2180,0800,0084,71DD  ;2737  SC_K[.14]
U 11DD, 0000,003C,0980,0800,0084,B1DE  ;2738  SC_SC-K[.2]
U 11DE, 0801,0036,0180,0800,0000,0001  ;2739  ALU_D.AND.MASK,D_ALU,RETURN1
            ;2740
```

```
            ;2741   .PAGE
            ;2742   .TOC "   SILCLR ROUTINE"
            ;2743 ;*****************************************************************
            ;2744 ;*
            ;2745 ; THIS ROUTINE UNLOCKS THE SILO,AND WAITS FOR 16 CYCLES
            ;2746 ; TO CLEAR IT OUT.
            ;2747 ;-
            ;2748 ;*****************************************************************
U 11DF, 0F00,003C,0180,0800,0000,11E0   ;2749 SILCLR: D_0
U 11E0, 0000,003C,7180,3C00,0000,11E1   ;2750  ID[COMP]_D
U 11E1, 0818,0038,1180,0800,0000,11E2   ;2751  D_K[.4]
U 11E2, 0001,003C,0180,0800,0010,11E3   ;2752 CLR0: ALU_D,CLK.UBCC
U 11E3, 0000,013C,0180,0800,0000,1084   ;2753  Z?
U 1084, 0819,0000,0580,0800,0000,11E2   ;2754 =0 J/CLR0,D_D-K[.1]
U 1085, 0000,003E,0180,0800,0000,0001   ;2755  RETURN1
            ;2756
```

```
            ;2757  .PAGE
            ;2758  .TOC "    SLOLCK ROUTINE"
            ;2759 ;*****************************************************************
            ;2760 ;*
            ;2761 ;       THIS ROUTINE PLACES THE LAST 15 LOCATIONS OF THE SILO
            ;2762 ; IN RC[0E] THRU RC[0].
            ;2763 ;
            ;2764 ;  THE STATE IN WHICH THE CALL IS DONE, AND
            ;2765 ;  THE PRECEDING STATE MUST LOOK EXACTLY AS FOLLOWS:
            ;2766 ;
            ;2767 ;   ID[COMP]_D,...
            ;2768 ;   CALL,J/SLOLCK,MCT/XXXXX
            ;2769 ;
            ;2770 ;   WHERE XXXXX IS ANY REFERENCE.
            ;2771 ;
            ;2772 ;-
            ;2773 ;*****************************************************************
            ;2774 =0
U 1086, 0818,0038,1180,0800,0000,1087  ;2775 SLOLCK: D_K[.4]    ; PROCEED TO WAIT A TOTAL OF 16 CYCLES
U 1087, 0001,003C,0180,0800,0010,11E4  ;2776 L1: ALU_D,CLK.UBCC  ; ...
U 11E4, 0819,0100,0580,0800,0000,1088  ;2777  Z?,D_D-K[.1]
U 1088, 0000,003C,0180,0800,0000,1087  ;2778 =0 J/L1    ; ...
U 1089, 0000,003C,6180,0800,0084,71E5  ;2779  SC_K[.F]   ; START FOR PLACING SILO IN RC REGISTERS
U 11E5, 0000,003C,0580,0800,0084,B1E6  ;2780 L2: SC_SC-K[.1]  ; STARTING WITH RC[0E]
U 11E6, 0000,003C,61F0,2C00,0000,11E7  ;2781  Q_ID[SILO]
U 11E7, 0001,203C,0180,0838,0000,11E8  ;2782  RC(SC)_ALU,ALU_Q
U 11E8, 0018,0038,1D80,0800,0010,11E9  ;2783  ALU_SC,CLK.UBCC
U 11E9, 0000,013C,0180,0800,0000,108A  ;2784  Z?
U 108A, 0000,003C,0180,0800,0000,11E5  ;2785 =0 J/L2
U 108B, 0F00,003C,0180,0800,0000,11EA  ;2786  D_0
U 11EA, 0000,003E,7180,3C00,0000,0001  ;2787  ID[COMP]_D,RETURN1 ; UNLOCK SILO
            ;2788
```

```
            ;2789  .PAGE
            ;2790  .TOC "   CLCPTO ROUTINE"
            ;2791 ;*****************************************************************
            ;2792 ;* THIS ROUTINE CLEARS THE SBI.ERR REGISTER
            ;2793 ;-
U 11EB, 081B,0000,0580,0800,0000,11EC  ;2794 CLCPTO: D_0-K[.1]
U 11EC, 0801,0028,6580,3C00,0000,11ED  ;2795  ID[SBI.ERR]_D,D_NOT.D
U 11ED, 0000,003E,6580,3C00,0000,0001  ;2796  ID[SBI.ERR]_D,RETURN1
            ;2797
```

```
            ;2798  .PAGE
            ;2799  .TOC "    SETMRG ROUTINE"
            ;2800 ;-
            ;2801 ;////////////////////////////////////////////////////////////////
            ;2802 ;*
            ;2803 ; THIS ROUTINE SETS UP THE MAINTENANCE REGISTER
            ;2804 ; DATA TO DISABLE CACHE AND RC0 <4:0> ARE PUT
            ;2805 ; IN BITS <27:23> (MAINTENANCE ID). THE DATA
            ;2806 ; IS RETURNED IN THE D REGISTER. WHEN THIS ROUTINE
            ;2807 ; IS CALLED RC0 <4:0> MUST CONTAIN THE ID TO BE
            ;2808 ; USED IN THE MAINTENANCE ID.
            ;2809 ;
U 11EE, 0810,0038,0180,0900,0000,11EF  ;2810 SETMRG: D_RC[0]
U 11EF, 0018,0038,7DC0,0800,0000,11F0  ;2811  Q_K[.18]
U 11F0, 0019,2000,09F8,0800,0082,11F1  ;2812  SC_Q-K[.2],Q_0
U 11F1, 0D00,003C,0180,0800,0000,11F2  ;2813  D_DAL.SC    ;SHIFT ID INTO BITS <27:23>
U 11F2, 0819,0030,4580,0800,0000,11F3  ;2814  D_D.OR.K[.8000]
U 11F3, 0500,003C,0180,0800,0000,11F4  ;2815  D_D.LEFT    ;DISABLE CACHE
U 11F4, 0819,0032,4580,0800,0000,0001  ;2816  D_D.OR.K[.8000],RETURN1
            ;2817
```

```
            ;2818  .PAGE
            ;2819  .TOC "   INIT ROUTINE"
            ;2820  ;=============================================================================
            ;2821  ;
            ;2822  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2823  ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2824  ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2825  ;
            ;2826  ;=============================================================================
            ;2827  ;
U 11F5, 0003,003C,0180,0AF8,0000,108C  ;2828 INIT: R[0F]_0     ;CLEAR U-TRAPS
U 108C, 0000,003D,0180,0800,0000,11DB  ;2829 =0 CALL[CLRSBI]   ; EXECUTE AN UNJAM SEQUENCE
U 108D, 0000,003C,0180,0800,0000,108E  ;2830  NOP
U 108E, 0000,003D,0180,0800,0000,1204  ;2831 =0 CALL[SETPSL]   ;SET PSL AND CLEAR SOFTWARE
            ;2832       ;  INTRUPT REG
U 108F, 0000,003C,0180,0800,0000,1090  ;2833  NOP
U 1090, 0000,003D,0180,0800,0000,120B  ;2834 =0 CALL[CLRFLT]   ;CLEAR CPU FAULT BIT
U 1091, 0000,003C,0180,0800,0000,1092  ;2835  NOP
U 1092, 0000,003D,0180,0800,0000,120F  ;2836 =0 CALL[CLRECC]   ;CLEAR ANY PENDING ECC INTRUPTS
U 1093, 0000,003C,0180,0800,0000,1094  ;2837  NOP
U 1094, 0000,003D,0180,0800,0000,1207  ;2838 =0 CALL[CLRTIM]   ;CLEAR TIMEOUT BIT
U 1095, 0000,003C,0180,0800,0000,1096  ;2839  NOP
U 1096, 0000,003D,0180,0800,0000,1213  ;2840 =0 CALL[ENBECC]   ;ENABLE ECC INTRUPTS
U 1097, 0000,003C,0180,0800,0000,1098  ;2841  NOP
U 1098, 0000,003D,0180,0800,0000,1167  ;2842 =0 CALL[ENBFLT]   ;ENBALE FAULT INTERRUPTS
U 1099, 0818,0038,4580,0800,0000,109A  ;2843  D_K[.8000]   ;ENABLE CACHE PARITY ERROR
            ;2844 =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 109A, 0000,003D,7980,3C00,0000,1189  ;2845   ID[PARITY]_D
U 109B, 0F00,003C,0180,0800,0000,11F6  ;2846  D_0
U 11F6, 0000,003C,4980,3C00,0000,11F7  ;2847  ID[TBER0]_D   ;SHUT TB OFF
U 11F7, 0000,003E,7180,3C00,0000,0001  ;2848  ID[COMP]_D,RETURN1  ;DISABLE SILO INTERRUPTS
            ;2849
```

```
            ;2850   .PAGE
            ;2851   .TOC "    CHECK FOR INTERRUPT VECTORS"
            ;2852   ;
            ;2853   ;*****************************************************************
            ;2854   ;+
            ;2855   ;CHECK FOR INTERRUPTS PENDING
            ;2856   ;
            ;2857   ; INTERRUPT  VECTOR CHECKED
            ;2858   ; ------------------------------------
            ;2859   ; WRITE TIMEOUT  ^X60
            ;2860   ; SBI FAULT  ^X5C
            ;2861   ; CRD,RDS TAG  ^X54
            ;2862   ; SILO    ^X50
            ;2863   ;
            ;2864   ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2865   ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2866   ;
            ;2867   ; DATA: EXPECTED VECTOR
            ;2868   ;  VECTOR STROBED
            ;2869   ;-
            ;2870   ;*****************************************************************
            ;2871   ;
            ;2872 SBIFLT: ;*****************************************************************
            ;2873   ;CHECKS FOR THE SBI FAULT INTERRUPT, ^X5C
            ;2874   ;*****************************************************************
            ;2875   ;
U 11F8, 0818,0038,3580,0800,4000,11F9   ;2876   IEK/ISTR,D_K[.50]   ;STROBE THE VECTOR
U 11F9, 0819,0030,8580,0800,0000,11FA   ;2877   D_D.OR.K[.C]    ;GENERATE THE EXPECTED VECTOR
U 11FA, 0000,003C,35F0,2C00,0000,11FB   ;2878   Q_ID[VECTOR]
U 11FB, 0019,2034,81C0,0800,0000,11FC   ;2879   Q_Q.AND.K[.3FF]
U 11FC, 001D,0020,0180,0800,0010,11FD   ;2880   ALU_D[XOR]Q,CLK.UBCC
U 11FD, 0000,013C,0180,0800,0000,109C   ;2881   Z?
U 109C, 0000,003C,0180,0800,0000,1175   ;2882  =0 J/ERRFLT    ;NOT THE EXPECTED INTERRUPT
U 109D, 0000,003E,0180,0800,0000,0002   ;2883   RETURN2
            ;2884 SILINT: ;*****************************************************************
            ;2885   ;CHECK FOR SILO INTERRUPT VECTOR    ^X50
            ;2886   ;*****************************************************************
            ;2887   ;
U 11FE, 0000,003C,0180,0800,4000,11FF   ;2888   IEK/ISTR    ;STROBE THE VECTOR
U 11FF, 0818,0038,3580,0800,0000,1200   ;2889   D_K[.50]
U 1200, 0000,003C,35F0,2C00,0000,1201   ;2890   Q_ID[VECTOR]
U 1201, 0019,2034,81C0,0800,0000,1202   ;2891   Q_Q.AND.K[.3FF]    ;MASK
U 1202, 001D,0020,0180,0800,0010,1203   ;2892   ALU_D[XOR]Q,CLK.UBCC   ;COMPARE
U 1203, 0000,013C,0180,0800,0000,109E   ;2893   Z?
U 109E, 0000,003C,0180,0800,0000,1175   ;2894  =0 J/ERRFLT    ;WRONG VECTOR
U 109F, 0000,003E,0180,0800,0000,0002   ;2895   RETURN2     ;O.K.,NORMAL RETURN
            ;2896
```

```
            ;2897   .PAGE
            ;2898   .TOC "    SETPSL ROUTINE"
            ;2899 SETPSL: ;=================================================================================
            ;2900   ;
            ;2901   ;DROP THE CPU IPR LEVEL TO
            ;2902   ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2903   ; PENDING
            ;2904   ;
            ;2905   ;=================================================================================
            ;2906   ;
U 1204. 0F00.003C.0180.0800.0000.1205   ;2907   D_0
U 1205. 0000.003C.3980.3C00.0000.1206   ;2908   ID[SIR]_D   ;SET SOFTWARE LEVELS TO ZERO
U 1206. 0000.003E.3D80.3C00.0000.0001   ;2909   RETURN1,ID[PSL]_D
            ;2910
```

```
            ;2911   .PAGE
            ;2912   .TOC "    CLRTIM ROUTINE"
            ;2913 CLRTIM: ;=================================================================
            ;2914   ;
            ;2915   ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2916   ; SBI ERROR REGISTER
            ;2917   ;
            ;2918   ;=================================================================
            ;2919   ;
U 1207,  0818.0038,0580,0800,0000,1208  ;2920  D_K[.1]
U 1208,  0000.003C,85F8,0800,0084,7209  ;2921  Q_0,SC_K[.C]
U 1209,  0D00.003C,0180,0800,0000,120A  ;2922  D_DAL,SC
U 120A,  0000.003E,6580,3C00,0000,0001  ;2923  ID[SBI.ERR]_D,RETURN1
            ;2924
```

```
            ;2925  .PAGE
            ;2926  .TOC "   CLRFLT ROUTINE"
            ;2927 CLRFLT: ;=================================================================
            ;2928  ;
            ;2929  ;CLEAR THE CP FAULT BIT IN THE
            ;2930  ; FAULT REGISTER
            ;2931  ;
            ;2932  ;=================================================================
            ;2933  ;
U 120B. 0818,0038,0180,0800,0000,120C  ;2934  D_K[.8]
U 120C. 0000,003C,65F8,0800,0084,720D  ;2935  Q_0,SC_K[.10]
U 120D. 0D00,003C,0180,0800,0000,120E  ;2936  D_DAL.SC
U 120E. 0000,003E,6D80,3C00,0000,0001  ;2937  ID[FAULT]_D,RETURN1
            ;2938
```

```
            ;2939   .PAGE
            ;2940   .TOC "    CLRECC ROUTINE"
            ;2941 CLRECC: ;=================================================================
            ;2942   ;
            ;2943   ;CLEAR CRD,RDS BIT IN THE
            ;2944   ;SBI ERROR REGISTER
            ;2945   ;
            ;2946   ;=================================================================
            ;2947   ;
U 120F,  0818,0038,0D80,0800,0000,1210  ;2948  D_K[.3]
U 1210,  0000,003C,89F8,0800,0084,7211  ;2949  Q_0,SC_K[.D]
U 1211,  0D00,003C,0180,0800,0000,1212  ;2950  D_DAL,SC
U 1212,  0000,003E,6580,3C00,0000,0001  ;2951  ID[SBI.ERR]_D,RETURN1
            ;2952
```

```
            ;2953  .PAGE
            ;2954  .TOC "   ENBECC ROUTINE"
            ;2955 ENBECC: ;=================================================================
            ;2956  ;
            ;2957  ;ENABLE CRD,RDS INTERRUPTS
            ;2958  ;
            ;2959  ;=================================================================
            ;2960  ;
U 1213. 0818,0038,0580,0800,0000,1214  ;2961  D_K[.1]
U 1214. 0000,003C,61F8,0800,0084,7215  ;2962  Q_0,SC_K[.F]
U 1215. 0D00,003C,0180,0800,0000,1216  ;2963  D_DAL,SC
U 1216. 0000,003E,6580,3C00,0000,0001  ;2964  ID[SBI.ERR]_D,RETURN1
            ;2965
```

```
            ;2966  .PAGE
            ;2967  .TOC "   Wait Loop Routine"
            ;2968 ;++
            ;2969 ; FUNCTIONAL DESCRIPTION:
            ;2970 ;
            ;2971 ; This routine is used to wait the specified number of machine cycles.
            ;2972 ; This routine is typically called to wait for an SBI write to
            ;2973 ; I/O space to complete.
            ;2974 ;
            ;2975 ; CALLING CONSTRAINT:
            ;2976 ;
            ;2977 ; =0
            ;2978 ;
            ;2979 ; INPUTS:
            ;2980 ;
            ;2981 ; d - Contains number of cycles to wait
            ;2982 ; alu.z condition code must not be set
            ;2983 ;
            ;2984 ; SIDE EFFECTS:
            ;2985 ;
            ;2986 ; d is destroyed
            ;2987 ;--
            ;2988 WAIT.LOOP:
U 1217, 0018,0038,6180,0800,0010,10A0  ;2989  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2990      ; ...is not set
            ;2991 =0
            ;2992 W.LOOP:
            ;2993  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10A0, 0819,0100,0580,0800,0010,10A0  ;2994       j/W.LOOP
U 10A1, 0000,003E,0180,0800,0000,0001  ;2995  return1    ; exit
            ;2996
```

```
            ;2997  .PAGE
            ;2998  .TOC "   RESET SUBTEST NUMBER"
            ;2999 ;++
            ;3000 ; FUNCTIONAL DESCRIPTION:
            ;3001 ;
            ;3002 ; Since some  of the tests will have to be restarted when two
            ;3003 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;3004 ; the subtest counter to zero ( only for thoes tests that have
            ;3005 ; more than one subtest). This subroutine may also be necessary
            ;3006 ; when a test is being loop on.
            ;3007 ;
            ;3008 ; CALLING CONSTRAINT:
            ;3009 ;
            ;3010 ; =0
            ;3011 ; SIDE EFFECTS:
            ;3012 ;
            ;3013 ;
            ;3014 ;--
            ;3015 RESET.SUBTST:
            ;3016  RC[0F]_0,    ; Reset subtest number
U 1218, 0003,003E,0180,09F8,0000,0001  ;3017   RETURN1    ; ...and return
            ;3018
```

```
            ;3019  .PAGE
            ;3020  .TOC "    WAIT REFRESH SUBROUTINE"
            ;3021 ;*************************************************************
            ;3022 ;**
            ;3023 ;
            ;3024 ; Functional Description:
            ;3025 ; ----------------------
            ;3026 ;
            ;3027 ; This subroutine provides the means of synchronizing the
            ;3028 ; diagnostic with the end of a refresh cycle on a MS780-E/H
            ;3029 ; memory.  It should be used by all the tests in which
            ;3030 ; timing is critical and delays due to refresh could
            ;3031 ; introduce false errors.
            ;3032 ;
            ;3033 ; Calling Constraint: =0
            ;3034 ; ------------------
            ;3035 ;
            ;3036 ;
            ;3037 ; Inputs:
            ;3038 ; ------
            ;3039 ;
            ;3040 ; RC[0E] must hold the I/O base address of the MS780-E/H
            ;3041 ;
            ;3042 ;
            ;3043 ; Outputs:
            ;3044 ; -------
            ;3045 ;
            ;3046 ; RETURN1 at the end of a refresh cycle.
            ;3047 ;
            ;3048 ;
            ;3049 ; Side Effects:
            ;3050 ; ------------
            ;3051 ;
            ;3052 ; The contents of the following registers will be lost:
            ;3053 ;
            ;3054 ; D, SC, CNFG Reg B
            ;3055 ;
            ;3056 ;--
            ;3057 ;*************************************************************
            ;3058 WAITREFRESH:
U 1219, 0010,0038,0180,0970,0200,121A  ;3059  VA_RC[0E]  ; Get address of CNFG Reg A
            ;3060  SC_K[.8],  ; Get data for CNFG  Reg B
U 121A, 0000,003C,0180,0803,0084,721B  ;3061   VA_VA+4  ; Point to CNFG Reg B
U 121B, 0803,001C,0180,0800,0000,121C  ;3062  D_NOT.MASK  ; Set bit (Refresh Lock Bit)
U 121C, 0000,003C,0180,A800,0000,121D  ;3063  CACHE.P_D[LONG]  ; Set bit 8 in CNFG Reg B
U 121D, 0F00,003C,0180,0800,0000,121E  ;3064  D_0  ; Get ready to clear the bit
U 121E, 0000,003C,0180,A800,0000,121F  ;3065  CACHE.P_D[LONG]  ; Clear bit 8 in CNFG Reg B
            ;3066 REFRESH.WAIT.LOOP:
            ;3067  D[LONG]_CACHE.P, ; Get contents of CNFG Reg B
U 121F, 0000,003C,0180,C800,0084,7220  ;3068   SC_K[.8] ; Point to bit 8
            ;3069  ALU_D.ANDNOT.MASK, ; Is bit 8 still set?
U 1220, 0001,0024,0180,0800,0010,1221  ;3070   CLK.UBCC ; ...
U 1221, 0000,013C,0180,0800,0000,10A2  ;3071  Z?  ;
U 10A2, 0000,003C,0180,0800,0000,121F  ;3072 =0 J/REFRESH.WAIT.LOOP ; Bit 8 is still set
U 10A3, 0818,0038,1180,0800,0000,10A4  ;3073  D_K[.4]  ; Number fof cycles to wait for
```

```
            ;3074      ; ...refresh to end
U 10A4, 0000,003D,0180,0800,0000,1217  ;3075 =0 CALL[WAIT.LOOP]  ; Go and wait for refresh to end
U 10A5, 0000,003E,0180,0800,0000,0001  ;3076  RETURN1   ; OK. Refresh should be done by now
            ;3077      ; ...Return to caller
            ;3078
            ;3079
            ;3080
```

```
            ;3081  .PAGE "TEST 199 SILO INTEGRITY TEST"
            ;3082 ;*****************************************************************
            ;3083 ;**
            ;3084 ;
            ;3085 ;TEST 196 SILO INTEGRITY TEST
            ;3086 ;
            ;3087 ;TEST DESCRIPTION
            ;3088 ; THIS TEST CHECKS THE INTEGRITY OF THE SILO BITS <29:25>
            ;3089 ; BY DOING A WRITE REFERENCE WHILE FORCE MULTIPLE TRANSMITTERS
            ;3090 ; IS SET IN THE MAINTENANCE REGISTER. THE SILO IS THEN CHECKED
            ;3091 ; FOR THE CORRECT ID.
            ;3092 ; TEST PATTERNS
            ;3093 ;      00001
            ;3094 ;      00010
            ;3095 ;      00100
            ;3096 ;      01000
            ;3097 ;      10000
            ;3098 ;
            ;3099 ;LOGIC DESCRIPTION
            ;3100 ; THIS TEST CHECKS THE SILO
            ;3101 ;
            ;3102 ;ERROR DESCRIPTION
            ;3103 ; EXPECTED WRITE DATA AS WOULD APPEAR IN SILO
            ;3104 ; RECEIVED WRITE DATA AS WOULD APPEAR IN SILO
            ;3105 ; ID UNDER TEST <4:0>
            ;3106 ;
            ;3107 ;--
            ;3108 ;*****************************************************************
            ;3109 =0
U 10A6,  0000.003D.0180.0800.0000.1145  ;3110 SLTST: NEWTST
U 10A7,  0000.003C.0180.0800.0000.1222  ;3111  NOP
U 1222,  0018.0038.0D80.09F8.0000.1223  ;3112  RC[0F]_K[.3]
U 1223,  0018.0038.D5C0.0800.0000.1224  ;3113  Q_K[.6]
U 1224,  0019.2000.0580.0A90.0000.1225  ;3114  R[02]_Q-K[.1]   ;INITIALIZE LOOP COUNT
U 1225,  0818.0038.9D80.0800.0000.1226  ;3115  D_K[.7C]
U 1226,  0000.003C.0180.0800.0084.7227  ;3116  SC_K[.8]
U 1227,  0801.001C.0180.0800.0000.1228  ;3117  D_D.ORNOT.MASK
U 1228,  0000.003C.65F8.0800.0084.7229  ;3118  SC_K[.10],Q_0
U 1229,  0D00.003C.0180.0800.0000.122A  ;3119  D_DAL.SC
U 122A,  0001.003C.0180.0AA0.0000.122B  ;3120  R[04]_D    ;EXP WR DATA AS IN SILO
U 122B,  0018.0038.B980.0AA8.0000.10A8  ;3121  R[05]_K[.19]   ;BIT TO SET EXP ID AS IN SILO
U 10A8,  0000.003D.0180.0800.0000.107A  ;3122 =0 CALL[CPUCA]
U 10A9,  0B00.003C.0180.0800.0000.122C  ;3123  D_D.SWAP
U 122C,  0600.003C.0180.0800.0000.122D  ;3124  D_D.RIGHT
U 122D,  0819.0034.8D80.0800.0000.122E  ;3125  D_D.AND.K[.1F]
U 122E,  0001.003C.0180.0990.0000.122F  ;3126  RC[02]_D   ;SAVE CPU ID IN RC2 <4:0>
U 122F,  0818.0038.0580.0800.0000.1230  ;3127  D_K[.1]
U 1230,  0001.003C.0180.09E0.0000.10AA  ;3128 CHKNT: RC[0C]_D   ;ID TEST PATTERN
            ;3129 =0 D_0,
U 10AA,  0F00.003D.0180.0800.0000.11BA  ;3130   CALL[GENERATE.IO]
U 10AB,  0001.003C.0180.0800.0200.10AC  ;3131  VA_D    ;NONEXISTANT I/O ADDRESS
U 10AC,  0000.003D.0180.0800.0000.115B  ;3132 =0 ERLOOP
U 10AD,  0000.003C.8980.0800.0084.7231  ;3133  SC_K[.D]
U 1231,  0003.001C.0180.0AF8.0000.1232  ;3134  R[0F]_ALU,ALU_NOT.MASK  ;SET TIMEOUT TRAP FLAG
U 1232,  0000.003C.2180.0800.0084.7233  ;3135  SC_K[.14]
```

```
U 1233, 0000,003C,0580,0800,0084,B234  ;3136  SC_SC-K[.1]
U 1234, 0803,001C,0180,0800,0000,1235  ;3137  D_NOT.MASK
U 1235, 0001,003C,0180,0A98,0000,1236  ;3138  R[03]_D
U 1236, 0F00,003C,6D80,3C00,0000,1237  ;3139  ID[FAULT]_D,D_0    ;CLEAR THE FAULT REG
U 1237, 0000,003C,7580,3C00,0000,10AE  ;3140  ID[MAINT]_D    ;CLEAR THE MAINT REG
U 10AE, 0000,003D,0180,0800,0000,11DB  ;3141 =0 CALL[SBI.UNJAM]    ;UNJAM THE SBI
U 10AF, 0000,003C,0180,0800,0000,10B0  ;3142  NOP
U 10B0, 0000,003D,0180,0800,0000,11DF  ;3143 =0 CALL[SILCLR]    ;CLEAR THE SILO
U 10B1, 0810,0038,0180,0960,0000,1238  ;3144  D_RC[0C]    ;GET TEST ID
U 1238, 0010,0038,01C0,0910,0000,1239  ;3145  Q_RC[02]    ;GET CPU ID
U 1239, 001D,0000,0180,0800,0010,123A  ;3146  ALU_D-Q,CLK.UBCC
U 123A, 0001,013C,0180,0980,0000,10B2  ;3147  Z?,RC[0]_D    ;CPU ID=TEST ID?
U 10B2, 0018,0038,6180,0988,0000,10B4  ;3148 =0 RC[01]_K[.F],J/CONTIN  ;LOCATION OF DATA IN SILO
U 10B3, 0018,0038,6580,0988,0000,10B4  ;3149  RC[01]_K[.10]
        ;3150 =0
U 10B4, 0000,003D,0180,0800,0000,11EE  ;3151 CONTIN: CALL[SETMRG]    ;PUT TEST ID IN MAINT REG
U 10B5, 0018,0038,51C0,0800,0000,123B  ;3152  Q_K[.1E]
U 123B, 0019,2000,0980,0800,0082,123C  ;3153  SC_Q-K[.2]
U 123C, 0801,001C,0180,0800,0000,123D  ;3154  D_D.ORNOT.MASK
U 123D, 0000,003C,7580,3C00,0000,10B6  ;3155  ID[MAINT]_D    ;SET MULT XMITTERS
U 10B6, 0F00,003D,0180,0800,0000,1179  ;3156 =0 CALL,J/DCE,D_0
U 10B7, 0000,003C,65F8,0800,0084,723E  ;3157  SC_K[.10],Q_0
U 123E, 0D00,003C,0180,0800,0000,123F  ;3158  D_DAL.SC
U 123F, 0000,003C,ED80,0800,0084,7240  ;3159  SC_K[.1B]
U 1240, 0801,001C,0180,0800,0000,1241  ;3160  D_D.ORNOT.MASK    ;SET COMP REG DATA
U 1241, 0000,003C,7180,3C00,0000,1242  ;3161  ID[COMP]_D    ;SET LOCK ON ID AND E IN COUNT
U 1242, 0000,003C,0180,A800,0000,1243  ;3162  MCT/WRITE.P    ;DO THE REFERENCE
U 1243, 0010,0038,0180,0908,0082,1244  ;3163  SC_RC[01]
U 1244, 0000,003C,0580,0800,0084,B245  ;3164 NXSLO: SC_SC-K[.1]
U 1245, 0000,003C,61F0,2C00,0000,1246  ;3165  Q_ID[SILO]    ;READ THE SILO
U 1246, 0018,0038,1D80,0800,0010,1247  ;3166  ALU_SC,CLK.UBCC
U 1247, 0000,013C,0180,0800,0000,10B8  ;3167  Z?
U 10B8, 0000,003C,0180,0800,0000,1244  ;3168 =0 J/NXSLO
U 10B9, 0001,203C,0180,09E8,0000,1248  ;3169  RC[0D]_Q
U 1248, 0000,003C,0180,0A28,0082,1249  ;3170  SC_R[05]
U 1249, 0800,003C,0180,0A20,0000,124A  ;3171  D_R[04]
U 124A, 0801,001C,0180,0800,0000,124B  ;3172  D_D.ORNOT.MASK    ;SET EXPECTED DATA
U 124B, 001D,0000,0180,0800,0010,124C  ;3173  ALU_D-Q,CLK.UBCC
U 124C, 0001,013C,0180,09F0,0000,10BA  ;3174  Z?,RC[0E]_D
U 10BA, 0000,003D,0180,0800,0000,115E  ;3175 =0 ERROR2    ;WRITE DATA IN SILO INCORRECT
U 10BB, 0000,003C,01C0,0A28,0000,124D  ;3176  Q_R[05]
U 124D, 0019,2014,0580,0AA8,0000,124E  ;3177  R[05]_Q+K[.1]    ;NEXT EXPECTED ID IN SILO
U 124E, 0810,0038,0180,0960,0000,124F  ;3178  D_RC[0C]
U 124F, 0500,003C,0180,0800,0000,1250  ;3179  D_D.LEFT,SI/ZERO  ;NEXT TEST PATTERN
U 1250, 0000,003C,01C0,0A10,0000,1251  ;3180  Q_R[02]    ;GET LOOP COUNT
U 1251, 0019,2000,0580,0A90,0010,1252  ;3181  R[02]_Q-K[.1],CLK.UBCC  ;DECREMENT LOOP COUNT
U 1252, 0000,013C,0180,0800,0000,10BC  ;3182  Z?
U 10BC, 0000,003C,0180,0800,0000,1230  ;3183 =0 J/CHKNT
U 10BD, 0800,003C,0180,0A18,0000,1253  ;3184  D_R[03]
U 1253, 0F00,003C,6D80,3C00,0000,1254  ;3185  ID[FAULT]_D,D_0    ;CLEAR THE FAULT
U 1254, 0000,003C,7580,3C00,0000,1255  ;3186  ID[MAINT]_D    ;CLEAR MULT XMITTERS
U 1255, 0000,003C,0180,0800,0000,10BE  ;3187  NOP
        ;3188
```

```
;3189 .PAGE "TEST 19A CHECK SBI COMMANDS AND STALL USING SILO TEST 1"
;3190 ;////////////////////////////////////////////////////////////////////////////
;3191 ;
;3192 ;
;3193 ;  SBI004.MIC
;3194 ;
;3195 ;============================================================================
;3196 ;**
;3197 ; SBI COMMANDS AND STALL USING SILO
;3198 ;
;3199 ;
;3200 ; TEST DESCRIPTION
;3201 ;
;3202 ;       THIS TEST CHECK THE COMMANDS ISSUED BY THE
;3203 ; CPU SBI INTERFACE.  THE STALL TIMING IS CHECKED
;3204 ; ALSO.  THE ADDRESS BIT #29 IS CHANGE TO SEE THAT
;3205 ; THE FUNCTIONS ARE PROPER IF A I/O REFERENCE IS MADE.
;3206 ; THE FOLLOWING TABLE IS USED TO DESCRIBE THE DIFFERENT CPU/
;3207 ; SBI OPERATIONS:
;3208 ;
;3209 ; NOTE: EXT R EXTENDED READ OPERATION
;3210 ;  EXT W EXTENDED WRITE OPERATION
;3211 ;  IR INTERLOCK READ
;3212 ;  IW INTERLOCK WRITE
;3213 ;  MASK R MASKED READ OPERATION
;3214 ;  MASK W MASKED WRITE OPERATION
;3215 ;
;3216 ;  IT SHOULD BE NOTED THAT READ.V.WCHK OPERATIONS ARE
;3217 ;  TRANSFORMED INTO INTERLOCK OPERATION IF I/O SPACE IS
;3218 ;  ADDRESSED.  THE IMPLICT INTERLOCK FLIP/FLOP (IMPLICT)
;3219 ;  IS USED TO TRANSFORM THE NEXT WRITE TO AN INTERLOCK WRITE
;3220 ;  OPERATION.
;3221 ;
;3222 ; MCT/FUNCTION  NON I/O  I/O  COMMENT
;3223 ;    (ADDR #29=0) (ADDR #29=1)
;3224 ;
;3225 ; READ.P   EXT R  MASK R
;3226 ; READ.V.RCHK  EXT R  MASK R
;3227 ; READ.V.NOCHK  EXT R  MASK R
;3228 ; READ.V.WCHK  EXT R  IR SET IMPLICT IF GOOD
;3229 ;        DATA RETURNED
;3230 ;
;3231 ;  WRITE.P   MASK W  IW(IF IMPLICT),MASK W
;3232 ;  WRITE.V.NOCHK  MASK W  IW(IF IMPLICT),MASK W
;3233 ;  WRITE.V.WCHK  MASK W  IW(IF IMPLICT),MASK W
;3234 ;
;3235 ;  NOTE:  IW(ANY) WILL CLEAR IMPLICT
;3236 ;
;3237 ; ================================================
;3238 ;
;3239 ; SUBTEST 1 GENERATE CONSTANTS FIND CPU ID
;3240 ;
;3241 ;
;3242 ; FUNCTION   OPERATION POINTER
;3243 ;    ^^^
```

```
;3244  ; SUBTEST 2 READ.P  (0) ADDR #29=0,1
;3245  ;  READ.V.RCHK (1) EXTENDED READS
;3246  ;  READ.V.NOCHK (2) MASKED READS
;3247  ;  READ.V.WCHK (3) INTERLOCK READS
;3248  ;
;3249  ;
;3250  ;
;3251  ;
;3252  ; SUBTEST 3 WRITE.P  (0) ADDR #29=0,1
;3253  ;  WRITE.V.NOCHK (1) MASKED WRITES
;3254  ;  WRITE.V.WCHK (2)
;3255  ;
;3256  ;
;3257  ;
;3258  ; LOGIC DESCRIPTION
;3259  ;
;3260  ;    SILO LOGIC
;3261  ;    MCT DECODE ROMS
;3262  ;    IMPLIT FLIP/FLOP LOGIC
;3263  ;    MCT CONTROL LOGIC
;3264  ;
;3265  ;
;3266  ; ERROR LOGOUT
;3267  ;
;3268  ; DATA: I/O BASE ADDRESS OF MS780E/F CURRENTLY UNDER TEST
;3269  ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER USED
;3270  ;  EXPECTED DATA FROM SILO
;3271  ;  RECEIVED DATA FROM SILO
;3272  ;  SILO COUNT, 0=1ST LOCATION AFTER LOCK
;3273  ;  OPERATION POINTER (OR ADDRESS FLAG)
;3274  ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3275  ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3276  ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3277  ;  INTERRUPT FLAG:
;3278  ;   0=>NO INTERRUPT OCCURRED
;3279  ;   1=>AN INTERRUPT OCCURRED
;3280  ;  ADDRESS FLAG (0=NON I/O, 1=I/O ADDRESS)
;3281  ;
;3282  ; SEE SUBTEST FOR FULL DESCRIPTION
;3283  ;
;3284  ;--
;3285  ;=================================================================================
;3286  ;   SCRATCH PADS
;3287  ; RA       II IIII = CPU I.D.
;3288  ; 0 C/A INTERLOCK WR  00II III0 1101 1100  0-0-0-0
;3289  ; 1 C/A INTERLOCK READ  00II III0 1101 0000  0-0-0-0
;3290  ; 2 C/A EXT READ   00II III0 1110 0000  0-0-0-0
;3291  ; 3 C/A MASKED READ   00II III0 1100 0100  0-0-0-0
;3292  ; 4 C/A MASKED WRITE  00II III0 1100 1000  0-0-0-0
;3293  ; 5 C/A EXTENDED WRITE  00II III0 1110 1100  0-0-0-0
;3294  ; 6 READ DATA   00II III0 0000 0000  0-0-0-0
;3295  ; 7 WRITE DATA (2ND FOR EXT W) 00II III1 0100 0000  0-0-0-0
;3296  ; 8 WRITE DATA (1ST FOR EXT W) 00II III1 0111 1100  0-0-0-0
;3297  ; 9 INTERLOCK MASK( SILO )  0100 0000 0000 0000  0-0-0-0
;3298  ; A CPU ARB (ARBITRITION)       0-0-0-0  <-TR->
```

```
            ;3299 ; B UNCONDITIONAL LOCK OFF 16 TIC   ^X 6000 0000
            ;3300 ; C SILO ID CONSTANT  00II III0 0000 0000  0-0-0-0
            ;3301 ; D CONFORMATION (ACK)   0-0-0  0001  0-0-0-0
            ;3302 ; E
            ;3303 ; F U-TRAP FLAG
            ;3304 ;
            ;3305 ;
            ;3306 ; RC
            ;3307 ; 0 CONFIG REG A ADDRESS    ^X 2000 2000
            ;3308 ; 1 CONFIG REG B ADDRESS    ^X 2000 2004
            ;3309 ; 2 CONFIG REG C ADDRESS    ^X 2000 2008
            ;3310 ; 3 CPU I.D.
            ;3311 ; 4 CPU ID SHIFTED TO MAINT ID POSITION
            ;3312 ; A ADDRESS FLAG
            ;3313 ; B OPERATION POINTER
            ;3314 ; C SILO COUNT
            ;3315 ; D DATA FROM SILO
            ;3316 ; E EXPECTED DATA
            ;3317 ; F # ARG TO CONSOLE
            ;3318 ;
            ;3319 ;================================================================================
            ;3320 ;
            ;3321 ;
            ;3322 ;
            ;3323 =0
U 10BE, 0000,003D,0180,0800,0000,1145  ;3324 STALL: NEWTST
            ;3325  rc[0F]_K[.A],   ;
U 10BF, 0F18,0038,F580,09F8,0000,1014  ;3326   D_0   ;
U 1014, 0000,003D,0180,0800,0000,1050  ;3327 =00 CALL[FIND.MS780E]  ; Find MS780-E/H on the SBI
U 1015, 0000,003D,1980,0800,0084,703E  ;3328  ERROR1[ZERO]   ; No MS780-E/H Found on the SBI
U 1016, 0000,003C,0180,0800,0000,1256  ;3329  NOP
            ;3330 =
            ;3331 RESTART.TEST:
U 1256, 001B,0000,8980,0800,0082,1257  ;3332  SC_ALU,ALU_0-K[.D]  ; Get -13 in SC
U 1257, 0810,0038,0180,0970,0000,1258  ;3333  D_rc[0E]   ; Get Memory I/O Base Address
U 1258, 0D00,003C,0180,0800,0000,1259  ;3334  D_DAL.SC   ; Shift I/O base address to
            ;3335       ; ...extract TR level
U 1259, 0819,0034,4980,0800,0000,125A  ;3336  D_D.AND.K[.FF]   ; Mask unwanted bits and keep only
            ;3337       ; ...TR level in D Reg
U 125A, 0000,003C,9180,3C00,0000,125B  ;3338  ID[24]_D   ; Save in ID Reg 24
U 125B, 0F00,003C,0180,0800,0000,125C  ;3339  D_0
U 125C, 0000,003C,7180,3C00,0000,125D  ;3340  ID[COMP]_D   ; CLEAR SILO COMPARATOR
U 125D, 0018,0038,0D80,09E8,0000,1018  ;3341  rc[0D]_K[.3]   ; Set flag for the fail-chain
U 1018, 0000,003D,0180,0800,0000,1004  ;3342 =00 CALL[SELECT.LOWER]  ; Try to enable Lower Controller
U 1019, 0000,003C,0180,0800,0000,101B  ;3343  J/TRY.UPPER   ; Lower ctrlr miscfg, try upper
U 101A, 0000,003C,0180,0800,0000,1022  ;3344  J/TEST.MEM   ; Lower configured, go test memory
            ;3345       ;
            ;3346 TRY.UPPER:
U 101B, 0018,0038,0D80,09E8,0000,1020  ;3347  RC[0D]_K[.3]   ; Set flag for failchain
U 1020, 0000,003D,0180,0800,0000,1010  ;3348 =00 CALL[SELECT.UPPER]  ; Try to enable upper cntrlr
            ;3349       ;
            ;3350 ;
            ;3351 ;////////////////////////////////////////////////////////////////////////////////
            ;3352 ;
U 1021, 0000,003D,0980,0800,0084,703E  ;3353  ERROR1[.2]   ; Could NOT configure either Controller
```

```
            ;3354 ;
            ;3355 ;////////////////////////////////////////////////////////////////////////////////
            ;3356 ;
            ;3357 ; ERROR LOGOUT:
            ;3358 ;
            ;3359 ; DATA:
            ;3360 ;  I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;3361 ;  ERROR CODE ( CONTROLLERS MISCONFIGURED )
            ;3362 ;
            ;3363 ;
            ;3364 ;////////////////////////////////////////////////////////////////////////////////
            ;3365 ;
            ;3366 TEST.MEM:
U 1022, 0810,0038,0180,0970,0000,1023  ;3367  D_RC[0E]   ; Get ready to offset I/O address to
            ;3368       ; Data latch #1
            ;3369  ALU_D+K[.10],   ; Add x10 to x20002000 for I/O address
U 1023, 0019,0014,6580,09E8,0000,10C0  ;3370  RC[0D]_ALU
            ;3371 ;
            ;3372 ;////////////////////////////////////////////////////////////////////////////////
            ;3373 ;++
            ;3374 ;
            ;3375 ; SUBTEST 1: CONSTANTS GENERATION
            ;3376 ;
            ;3377 ;--
            ;3378 ;////////////////////////////////////////////////////////////////////////////////
            ;3379 ;
            ;3380 =0
            ;3381 TEST.15:
U 10C0, 0018,0039,8580,09F8,0000,1162  ;3382  CALL[SUBTEST1],rc[0F]_K[.C]
U 10C1, 0F00,003C,ED80,0800,0084,725E  ;3383  D_0,SC_K[.1B]   ;RA 9
U 125E, 0000,003C,0D80,0800,0084,925F  ;3384  SC_SC+K[.3]
U 125F, 0801,001C,0180,0AC8,0000,1260  ;3385  R[9]_ALU,D_D.ORNOT.MASK
U 1260, 0000,003C,01E0,0800,0000,1261  ;3386  Q_D     ;RA B
U 1261, 0600,003C,0180,0800,0000,1262  ;3387  D_D.RIGHT
U 1262, 081D,0030,0180,0AD8,0000,1263  ;3388  R[0B]_ALU,D_D.OR.Q
            ;3389    ;=============================================================
            ;3390    ; FIND CPU ID
            ;3391    ;=============================================================
U 1263, 0003,003C,0180,0990,0000,10C2  ;3392  rc[2]_0    ;SET INTIAL CPU ID
U 10C2, 0000,003D,0180,0800,0000,115B  ;3393 =0 ERLOOP
U 10C3, 0000,003C,0180,0800,0000,10C4  ;3394 S05010: NOP
U 10C4, 0000,003D,0180,0800,0000,11F5  ;3395 =0 CALL[INIT]
U 10C5, 0F00,003C,7D80,0800,0084,7264  ;3396  SC_K[.18],D_0    ;GET FORCE MX BIT
U 1264, 0000,003C,1180,0800,0084,9265  ;3397  SC_SC+K[.4]
U 1265, 0801,001C,0180,0800,0000,1266  ;3398  D_D.ORNOT.MASK
U 1266, 0000,003C,75F0,2C00,0000,1267  ;3399  Q_ID[MAINT]    ;GET CACHE OFF BITS
U 1267, 081D,0030,0180,0800,0000,1268  ;3400  D_D[OR]Q    ;OR THE TWO TOGETHER
U 1268, 0000,003C,7580,3C00,0000,1269  ;3401  ID[MAINT]_D    ;SAVE IT IN DESTINATION
U 1269, 0810,0038,0180,0910,0000,126A  ;3402  D_rc[2]
U 126A, 0019,0020,6580,0800,0010,126B  ;3403  ALU_D[XOR]K[.10],CLK.UBCC
U 126B, 0819,0114,0580,0990,0000,10C6  ;3404  Z?,RC[2]_ALU,D_D+K[.1]  ;GET NEXT ID
U 10C6, 0000,003C,0180,0800,0000,126C  ;3405 =0 J/S05020
U 10C7, 0000,003D,0180,0800,0000,115C  ;3406  ERROR1     ;DIDN'T SEE A MULTIPLE XMITTER
            ;3407       ;AFTER ALL 16 ID'S CHECKED
U 126C, 0000,003C,21F8,0800,0084,726D  ;3408 S05020: SC_K[.14],Q_0    ;SHIFT TEST ID TO MAINT
```

```
U 126D. 0000.003C.0D80.0800.0084.926E  ;3409  SC_SC+K[.3]   ; ID POSITION
U 126E. 0D00.003C.0180.0800.0000.126F  ;3410  D_DAL.SC
U 126F. 0000.003C.75F0.2C00.0000.1270  ;3411  Q_ID[MAINT]
U 1270. 081D.0030.0180.0800.0000.1271  ;3412  D_D[OR]Q   ;GOT FORCE MX,ID AND
U 1271. 0000.003C.7580.3C00.0000.1272  ;3413  ID[MAINT]_D   ;  CACHE OFF
U 1272. 0018.0038.1980.0800.0200.1273  ;3414  VA_K[ZERO]
U 1273. 0000.003C.0180.A800.0000.10C8  ;3415  CACHE.P_D[LONG]   ;WRITE CAUSING MULTIPLE  *******
        ;3416      ; ...XMITTER FAULT
        ;3417 =0 D_K[.4],   ; WAIT 6 CLOCK TICKS
U 10C8. 0818.0039.1180.0800.0000.1217  ;3418   CALL[WAIT.LOOP]  ; ...
U 10C9. 0F00.003C.0180.0800.0000.1274  ;3419  D_0
U 1274. 0000.003C.7580.3C00.0000.1024  ;3420  ID[MAINT]_D   ;CLEAR FORCE FAULT
U 1024. 0000.003D.0180.0800.0000.11F8  ;3421 =00 CALL[SBIFLT]
U 1025. 0000.003C.0180.0800.0000.127E  ;3422  J/S05030   ;NO SBI FAULT,GOT ID
U 1026. 0818.0038.85F8.0800.0000.1027  ;3423  D_K[.C],Q_0   ;CHECK THE FAULT BITS SET
U 1027. 0000.003C.7D80.0800.0084.7275  ;3424  SC_K[.18]
U 1275. 0D00.003C.0180.0800.0000.1276  ;3425  D_DAL.SC
U 1276. 0001.003C.0180.09E0.0000.1277  ;3426  RC[0C]_D   ;SAVE EXPECTED DATA
U 1277. 0818.0038.F1F8.0800.0000.1278  ;3427  D_K[.FFFC],Q_0   ;GENERATE MASK FOR SBI FAULT BITS
U 1278. 0000.003C.7D80.0800.0084.7279  ;3428  SC_K[.18]
U 1279. 0D00.003C.0180.0800.0000.127A  ;3429  D_DAL.SC
U 127A. 0000.003C.6DF0.2C00.0000.127B  ;3430  Q_ID[FAULT]   ;GET FAULT REGISTER IMAGE
U 127B. 081D.0034.0180.09D8.0000.127C  ;3431  RC[0B]_ALU,D_ALU,ALU_D[AND]Q ;MASK AND SAVE
U 127C. 0011.0020.0180.0960.0010.127D  ;3432  ALU_D[XOR]RC[0C],CLK.UBCC ;ARE THE RIGHT BITS SET
U 127D. 0000.013C.0180.0800.0000.10CA  ;3433  Z?
U 10CA. 0000.003D.0180.0800.0000.115C  ;3434 =0 ERROR1     ;WRONG FAULT BITS SET
U 10CB. 0000.003C.0180.0800.0000.10C3  ;3435  J/S05010   ;GO TRY NEXT ID
U 127E. 0810.0038.01F8.0910.0000.127F  ;3436 S05030: D_rc[2],Q_0   ;SHIFT ID TO SBI SILO ID
U 127F. 0000.003C.7D80.0800.0084.7280  ;3437  SC_K[.18]   ; POSITION
U 1280. 0000.003C.0580.0800.0084.9281  ;3438  SC_SC+K[.1]
U 1281. 0D00.003C.0180.0800.0000.1282  ;3439  D_DAL.SC
U 1282. 0001.003C.0180.0998.0000.1283  ;3440  RC[3]_D     ;SAVE IT IN RC 3
U 1283. 0818.0038.89F8.0800.0000.1284  ;3441  D_K[.D],Q_0   ;RA 0
U 1284. 0000.003C.1180.0800.0084.7285  ;3442  SC_K[.4]
U 1285. 0D00.003C.0180.0800.0000.1286  ;3443  D_DAL.SC
U 1286. 0819.0030.8580.0800.0000.1287  ;3444  D_D.OR.K[.C]
U 1287. 0000.003C.6580.0800.0084.7288  ;3445  SC_K[.10]
U 1288. 0D00.003C.0180.0800.0000.1289  ;3446  D_DAL.SC
U 1289. 0811.0030.0180.0918.0000.128A  ;3447  D_ALU,ALU_D[OR]RC[3]
U 128A. 0001.003C.0180.0A80.0000.128B  ;3448  R[0]_D
U 128B. 0818.0038.89F8.0800.0000.128C  ;3449  D_K[.D],Q_0   ;RA 1
U 128C. 0000.003C.1180.0800.0084.928D  ;3450  SC_SC+K[.4]
U 128D. 0D00.003C.0180.0800.0000.128E  ;3451  D_DAL.SC
U 128E. 0811.0030.0180.0918.0000.128F  ;3452  D_ALU,ALU_D[OR]RC[3]
U 128F. 0F01.003C.0180.0A88.0000.10CC  ;3453  R[1]_D,D_0
U 10CC. 0000.003D.0180.0800.0000.1179  ;3454 =0 CALL[DCE]   ;RA 2
U 10CD. 0000.003C.6580.0800.0084.7290  ;3455  SC_K[.10]
U 1290. 0000.003C.1180.0800.0084.9291  ;3456  SC_SC+K[.4]
U 1291. 0D00.003C.0180.0800.0000.1292  ;3457  D_DAL.SC
U 1292. 0811.0030.0180.0918.0000.1293  ;3458  D_ALU,ALU_D[OR]RC[3]
U 1293. 0001.003C.0180.0A90.0000.1294  ;3459  R[2]_D
U 1294. 0818.0038.85F8.0800.0000.1295  ;3460  D_K[.C],Q_0   ;RA 3
U 1295. 0000.003C.1180.0800.0084.7296  ;3461  SC_K[.4]
U 1296. 0D00.003C.0180.0800.0000.1297  ;3462  D_DAL.SC
U 1297. 0819.0030.1180.0800.0000.1298  ;3463  D_D.OR.K[.4]
```

```
U 1298. 0000.003C.6580.0800.0084.7299  ;3464  SC_K[.10]
U 1299. 0D00.003C.0180.0800.0000.129A  ;3465  D_DAL.SC
U 129A. 0811.0030.0180.0918.0000.129B  ;3466  D_ALU,ALU_D[OR]RC[3]
U 129B. 0001.003C.0180.0A98.0000.129C  ;3467  R[3]_D
U 129C. 0818.0038.85F8.0800.0000.129D  ;3468  D_K[.C],Q_0    ;RA 4
U 129D. 0000.003C.1180.0800.0084.729E  ;3469  SC_K[.4]
U 129E. 0D00.003C.0180.0800.0000.129F  ;3470  D_DAL.SC
U 129F. 0819.0030.0180.0800.0000.12A0  ;3471  D_D.OR.K[.8]
U 12A0. 0000.003C.6580.0800.0084.72A1  ;3472  SC_K[.10]
U 12A1. 0D00.003C.0180.0800.0000.12A2  ;3473  D_DAL.SC
U 12A2. 0811.0030.0180.0918.0000.12A3  ;3474  D_ALU,ALU_D[OR]RC[3]
U 12A3. 0001.003C.0180.0AA0.0000.12A4  ;3475  R[4]_D
U 12A4. 0811.0030.0180.0918.0000.12A5  ;3476  D_ALU,ALU_D[OR]RC[3]  ;RA 5
U 12A5. 0000.003C.2180.0800.0084.72A6  ;3477  SC_K[.14]
U 12A6. 0000.003C.0580.0800.0084.92A7  ;3478  SC_SC+K[.1]    ;SET BIT #21
U 12A7. 0801.001C.0180.0AA8.0000.12A8  ;3479  R[5]_ALU,D_D.ORNOT.MASK
U 12A8. 0810.0038.0180.0918.0000.12A9  ;3480  D_RC[3]     ;RA 6
U 12A9. 0001.003C.0180.0AB0.0000.12AB  ;3481  R[6]_D
U 12AB. 0818.0038.F5F8.0800.0000.12AC  ;3482  D_K[.A],Q_0    ;RA 7
U 12AC. 0000.003C.2180.0800.0084.72AD  ;3483  SC_K[.14]
U 12AD. 0000.003C.0580.0800.0084.92AE  ;3484  SC_SC+K[.1]    ;SKIP 21 PLACES FOR LEFT
U 12AE. 0D00.003C.0180.0800.0000.12AF  ;3485  D_DAL.SC
U 12AF. 0811.0030.0180.0918.0000.12B0  ;3486  D_ALU,ALU_D[OR]RC[3]
U 12B0. 0001.003C.6580.0AB8.0084.72B1  ;3487  R[7]_D,SC_K[.10]
U 12B1. 0818.0038.61F8.0800.0000.12B2  ;3488  D_K[.F],Q_0    ;RA 8
U 12B2. 0000.003C.0980.0800.0084.92B3  ;3489  SC_SC+K[.2]
U 12B3. 0D00.003C.0180.0800.0000.12B4  ;3490  D_DAL.SC
U 12B4. 080D.0030.0180.0A38.0000.12B5  ;3491  D_ALU,ALU_D[OR]R[7]
U 12B5. 0001.003C.0180.0AC0.0000.12B6  ;3492  R[8]_D
U 12B6. 0F10.0038.0180.0910.0082.12B7  ;3493  SC_ALU,ALU_rc[2],D_0  ;RA A
U 12B7. 0801.001C.0180.0AD0.0000.12B8  ;3494  R[0A]_ALU,D_D.ORNOT.MASK
U 12B8. 0818.0038.4580.0800.0000.12B9  ;3495  D_K[.8000]    ;RA D
U 12B9. 0500.003C.0180.0800.0000.12BA  ;3496  D_D.LEFT
U 12BA. 0001.003C.0180.0AE8.0000.10CE  ;3497  R[0D]_D
        ;3498 ;=================================================================================
        ;3499 ;*
        ;3500 ;SUBTEST 2  READ OPERATIONS
        ;3501 ;
        ;3502 ; FUNCTION OPERATION POINTER
        ;3503 ; -------- ------------------------------------
        ;3504 ; READ.P    (0)  ADDR BIT #29=0,1
        ;3505 ; READ.V.RCHK  (1)
        ;3506 ; READ.V.NOCHK  (2)
        ;3507 ; READ.V.WCHK  (3)
        ;3508 ;
        ;3509 ; CHECKS FOR SILO INTERRUPT
        ;3510 ; SEE THE FOLLOWING IN SILO
        ;3511 ;
        ;3512 ; NULL (4 TICKS)
        ;3513 ; CPU ARB (IF I.D.NOT-10)
        ;3514 ; C/A (EXT READ,NON I/O) (MASKED,I/O SPACE) (INTERLOCK IF I/O AND
        ;3515 ;       OPERATION(3)  )
        ;3516 ; NULL
        ;3517 ; ACK
        ;3518 ; MEMORY ARB
```

```
            ;3519 ; READ DATA (HOLD) (SKIP IF MASKED READ (I/O))
            ;3520 ; READ DATA
            ;3521 ; ACK ( NULL IF I/O REFERENCE)
            ;3522 ; ACK
            ;3523 ; NULL(UNTILL COUNT = F)
            ;3524 ;
            ;3525 ;ERROR LOGOUT
            ;3526 ;
            ;3527 ; DATA: EXPECTED
            ;3528 ;  DATA FROM SILO
            ;3529 ;  POSITION READ IN SILO, 0=1ST POSITION AFTER LOCK
            ;3530 ;  OPERATION POINTER (POINTS TO MCT FUNCTION UNDER TEST)
            ;3531 ;    I.E. INDICATED BY (X)
            ;3532 ;  ADDRESS FLAG (0=NON I/O, 1=I/O REFERENCE)
            ;3533 ;  TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
            ;3534 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;3535 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;3536 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;3537 ;  INTERRUPT FLAG:
            ;3538 ;   0=>NO INTERRUPT OCCURRED
            ;3539 ;   1=>AN INTERRUPT OCCURRED
            ;3540 ;
            ;3541 ;-
            ;3542 ;================================================================================
            ;3543 ;
U 10CE, 0000,003D,0180,0800,0000,1160  ;3544 =0 SUBTEST
U 10CF, 0003,003C,0180,09C0,0000,12BB  ;3545  RC[8]_0    ;SET ADDRESS FLAG
U 12BB, 0003,003C,0180,09C8,0000,10D0  ;3546 S05100: RC[9]_0    ;RESET OPERATION POINTER
U 10D0, 0000,003D,0180,0800,0000,115B  ;3547 =0 ERLOOP
U 10D1, 0000,003C,0180,0800,0000,10D2  ;3548  NOP
            ;3549 =0
            ;3550 RETRY1:
U 10D2, 0000,003D,0180,0800,0000,11F5  ;3551 S05105: CALL[INIT]
U 10D3, 0010,0038,0180,0970,0200,12BC  ;3552  VA_RC[0E]    ; Load VA with CNFG Reg. A address
U 12BC, 0000,003C,0180,C800,0000,10D4  ;3553  D[LONG]_CACHE.P    ; Read it to clear any Fault bits
U 10D4, 0000,003D,0180,0800,0000,1219  ;3554 =0 CALL[WAITREFRESH]
U 10D5, 0010,0038,0180,0940,0010,12BD  ;3555  ALU_RC[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 12BD, 0018,0138,1980,0800,0200,10D6  ;3556  Z?,VA_K[ZERO]    ;NON I/O REFERENCE
U 10D6, 0010,0038,0180,0968,0200,10D7  ;3557 =0 VA_RC[0D]    ;I/O REFERENCE
U 10D7, 0800,003C,0180,0A58,0000,12BE  ;3558  D_R[0B]    ;ENABLE SILO
U 12BE, 0810,0038,7180,3D48,0000,12BF  ;3559  ID[COMP]_D,D_RC[9]  ;GET OPERATION POINTER
U 12BF, 0000,193C,0180,0800,0000,101C  ;3560  D1-0?    ;BRANCH TO OPERATION
U 101C, 0000,003C,0180,C800,0000,10D8  ;3561 =1100 MCT/READ.P,J/S05110  ; (0)
U 101D, 0000,003C,0180,4000,0000,10D8  ;3562  MCT/READ.V.RCHK,J/S05110 ; (1)
U 101E, 0000,003C,0180,4800,0000,10D8  ;3563  MCT/READ.V.NOCHK,J/S05110 ; (2)
U 101F, 0000,003C,0180,5000,0000,10D8  ;3564  MCT/READ.V.WCHK,J/S05110 ; (3)
            ;3565 =0
            ;3566 S05110: D_K[.14],   ; WAIT FOR THE SILO TO
U 10D8, 081B,0039,2180,0800,0000,1217  ;3567   CALL[WAIT.LOOP]  ; ...FILL UP ABOUT 24 CLOCK TICKS
U 10D9, 0000,003C,0180,0800,0000,1028  ;3568  NOP
U 1028, 0000,003D,0180,0800,0000,11FE  ;3569 =00 CALL[SILINT]
U 1029, 0000,003D,0180,0800,0000,115C  ;3570  ERROR1    ;DIDN'T GET SILO INTERRUPT
            ;3571    ;============================================================
            ;3572    ;
            ;3573    ; CHECK CONTENTS OF SILO
```

```
            ;3574   ;
            ;3575   ;=================================================================
U 102A. 0003.003C.0180.09D0.0000.102B  ;3576  RC[0A]_0   ;RESET SILO COUNTER
U 102B. 0003.003C.0180.09E0.0000.12C0  ;3577  RC[0C]_0   ;SET EXPECTED DATA
            ;3578       ;=====================================
U 12C0. 0000.003C.61F0.2C00.0000.12C1  ;3579 S05120: Q_ID[SILO]   ;SEE 4 NULL CYCLES
U 12C1. 0001.203C.0180.09D8.0010.12C2  ;3580  RC[0B]_Q,CLK.UBCC
U 12C2. 0810.0138.0180.0950.0000.10DA  ;3581  Z?,D_RC[0A]
U 10DA. 0000.003D.0180.0800.0000.115C  ;3582 =0 ERROR1    ;DIDN'T GET NULL CYCLE
U 10DB. 0019.0020.0D80.0800.0010.12C3  ;3583  ALU_D[XOR]K[.3],CLK.UBCC ;  NULL CYCLE?
U 12C3. 0000.013C.0180.0800.0000.10DC  ;3584  Z?
U 10DC. 0819.0014.0580.09D0.0000.12C0  ;3585 =0 RC[0A]_ALU,D_D+K[.1],J/S05120 ;NO, REPEAT LOOP
U 10DD. 0818.0038.6580.0800.0000.12C4  ;3586  D_K[.10]   ;CHECK CPU I.D.
U 12C4. 0011.0020.0180.0910.0010.12C5  ;3587  ALU_D[XOR]RC[2],CLK.UBCC ;
U 12C5. 0810.0138.0180.0950.0000.10DE  ;3588  Z?,D_RC[0A]
U 10DE. 0000.003C.0180.0800.0000.12C6  ;3589 =0 J/S05130   ;I.D. NOT 1,0,CHECK ARB
U 10DF. 0000.003C.0180.0800.0000.10E1  ;3590  J/S05140   ;I.D.=10,SKIP ARB
            ;3591   ;=================================================================
            ;3592   ;SEE CPU ARB IN SILO
            ;3593   ;=================================================================
U 12C6. 0819.0014.0580.09D0.0000.12C7  ;3594 S05130: RC[0A]_ALU,D_D+K[.1]  ;UPDATE COUNTER
U 12C7. 0800.003C.61F0.2E50.0000.12C8  ;3595  D_R[0A],Q_ID[SILO]  ;SET EXPECTED DATA
U 12C8. 0001.003C.0180.09E0.0000.12C9  ;3596  RC[0C]_D
U 12C9. 0001.203C.0180.09D8.0000.12CA  ;3597  RC[0B]_Q   ;SET DATA READ
U 12CA. 001D.0020.0180.0800.0010.12CB  ;3598  ALU_D[XOR]Q,CLK.UBCC
U 12CB. 0810.0138.0180.0950.0000.10E0  ;3599  Z?,D_RC[0A]
U 10E0. 0000.003D.0180.0800.0000.115C  ;3600 =0 ERROR1    ;DIDN'T SEE ARB IN SILO
            ;3601   ;=================================================================
            ;3602   ;SEE C/A FOR READ OPERATION
            ;3603   ;=================================================================
U 10E1. 0819.0014.0580.09D0.0000.12CC  ;3604 S05140: RC[0A]_ALU,D_D+K[.1]  ;SEE READ C/A
U 12CC. 0010.0038.0180.0940.0010.12CD  ;3605  ALU_RC[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 12CD. 0000.013C.0180.0800.0000.10E2  ;3606  Z?
U 10E2. 0800.003C.0180.0A08.0000.12CE  ;3607 =0 J/S05143,D_R[1]   ;I/O,SEE MASKED OR INTERLOCK
U 10E3. 0800.003C.0180.0A10.0000.10E5  ;3608  J/S05147,D_R[2]   ;NON I/O, SEE EXT READ
U 12CE. 0010.0038.01C0.0948.0000.12CF  ;3609 S05143: Q_RC[9]   ;I/O FLAG SET
U 12CF. 0019.2020.0D80.0800.0010.12D0  ;3610  ALU_Q.XOR.K[.3],CLK.UBCC ;
U 12D0. 0000.013C.0180.0800.0000.10E4  ;3611  Z?
U 10E4. 0800.003C.0180.0A18.0000.10E5  ;3612 =0 D_R[3]    ;NOT READ.V.WCHK,SEE MASKED READ
U 10E5. 0001.003C.61F0.2DE0.0000.12D1  ;3613 S05147: RC[0C]_D,Q_ID[SILO]  ;CHECK C/A SILO DATA
U 12D1. 0001.203C.0180.09D8.0000.12D2  ;3614  RC[0B]_Q
U 12D2. 001D.0020.0180.0800.0010.12D3  ;3615  ALU_D[XOR]Q,CLK.UBCC
U 12D3. 0810.0138.0180.0950.0000.10E6  ;3616  Z?,D_RC[0A]
U 10E6. 0000.003D.0180.0800.0000.115C  ;3617 =0 ERROR1    ;DIDN'T SEE READ C/A IN SILO
            ;3618   ;=================================================================
            ;3619   ;SEE NULL AFTER READ C/A(INTERLOCK SET IF I/O AND READ.V.WCHK)
            ;3620   ;=================================================================
U 10E7. 0819.0014.0580.09D0.0000.12D4  ;3621  RC[0A]_ALU,D_D+K[.1]  ;SEE NULL CYCLE
U 12D4. 0010.0038.01C0.0948.0000.12D5  ;3622  Q_RC[9]   ;IS  IT READ.V.WCHK ?
U 12D5. 0019.2020.0D80.0800.0010.12D6  ;3623  ALU_Q.XOR.K[.3],CLK.UBCC
U 12D6. 0000.013C.0180.0800.0000.10E8  ;3624  Z?
U 10E8. 0F00.003C.0180.0800.0000.10EB  ;3625 =0 D_0,J/S05149   ;NOT A READ.V.WCHK SEE NULL
U 10E9. 0010.0038.0180.0940.0010.12D7  ;3626  ALU_RC[8],CLK.UBCC  ;I/O REFERENCE ??
U 12D7. 0F00.013C.0180.0800.0000.10EA  ;3627  Z?,D_0    ;NON I/O REFERENCE
U 10EA. 0800.003C.0180.0A48.0000.10EB  ;3628 =0 D_R[9]    ;I/O REFERENCE AND READ.V.WCHK
```

```
U 10EB, 0001,003C,61F0,2DE0,0000,12D8  ;3629 S05149: RC[0C]_D,Q_ID[SILO]  ;SET EXPECTED DATA ,READ SILO
U 12D8, 001D,0020,0180,0800,0010,12D9  ;3630  ALU_D[XOR]Q,CLK.UBCC
U 12D9, 0810,0138,0180,0950,0000,10EC  ;3631  Z?,D_RC[0A]
U 10EC, 0000,003D,0180,0800,0000,115C  ;3632 =0 ERROR1     ;DIDN'T SEE NULL(INTERLOCK SET)
          ;3633       ;  IN SILO
          ;3634    ;================================================================
          ;3635    ;SEE ACK FROM MEMORY
          ;3636    ;================================================================
U 10ED, 0819,0014,0580,09D0,0000,12DA  ;3637  RC[0A]_ALU,D_D+K[.1]  ;SEE ACK FROM MEMORY
U 12DA, 0800,003C,61F0,2E68,0000,12DB  ;3638  D_R[0D],Q_ID[SILO]
U 12DB, 0001,003C,0180,09E0,0000,12DC  ;3639  RC[0C]_D
U 12DC, 0001,203C,0180,09D8,0000,12DD  ;3640  RC[0B]_Q
U 12DD, 001D,0034,01C0,0800,0000,12DE  ;3641  Q_D[AND]Q    ; Clear unwanted bits
U 12DE, 001D,0020,0180,0800,0010,12DF  ;3642  ALU_D[XOR]Q,CLK.UBCC
U 12DF, 0000,013C,0180,0800,0000,10EE  ;3643  Z?
U 10EE, 0000,003D,0180,0800,0000,115C  ;3644 =0 ERROR1     ;DIDN'T SEE ACK
          ;3645       ; FROM MEMORY
          ;3646    ;================================================================
          ;3647    ;SEE MEMORY ARB
          ;3648    ;================================================================
U 10EF, 0000,003C,91F0,2C00,0000,12E0  ;3649  Q_ID[24]    ; GET ID OF MEMORY
U 12E0, 0001,203C,0180,0800,0082,12E1  ;3650  SC_Q
U 12E1, 0803,001C,0180,09E0,0000,12E2  ;3651  RC[0C]_NOT.MASK,D_ALU  ; SET TR OF MEMORY
U 12E2, 0010,0038,01C0,0958,0000,12E3  ;3652  Q_RC[0B]    ; Get contents of SILO
U 12E3, 001D,0034,01C0,0800,0000,12E4  ;3653  Q_D[AND]Q    ; Mask unwanted bits
U 12E4, 001D,0020,0180,0800,0010,12E5  ;3654  ALU_D[XOR]Q,CLK.UBCC
U 12E5, 0810,0138,0180,0950,0000,10F0  ;3655  Z?,D_RC[0A]
U 10F0, 0000,003D,0180,0800,0000,115C  ;3656 =0 ERROR1     ;DIDN'T SEE ARB FROM MEMORY
U 10F1, 0010,0038,0180,0940,0010,12E6  ;3657  ALU_RC[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 12E6, 0000,013C,0180,0800,0000,10F2  ;3658  Z?     ;SKIP READ DATA AND HOLD
U 10F2, 0000,003C,0180,0800,0000,10F5  ;3659 =0 J/S05150    ; IF I/O REFERENCE
          ;3660    ;================================================================
          ;3661    ;SEE READ DATA AND HOLD
          ;3662    ;================================================================
U 10F3, 0819,0014,0580,09D0,0000,12E7  ;3663  RC[0A]_ALU,D_D+K[1]  ;SEE READ DATA AND HOLD
U 12E7, 0818,0030,0580,0A30,0000,12E8  ;3664  D_ALU,ALU_R[6].OR.K[.1]
U 12E8, 0001,003C,61F0,2DE0,0000,12E9  ;3665  RC[0C]_D,Q_ID[SILO]
U 12E9, 0003,003C,0180,09D8,0000,12EA  ;3666  RC[0B]_0
U 12EA, 001D,0020,0180,0800,0010,12EB  ;3667  ALU_D[XOR]Q,CLK.UBCC
U 12EB, 0810,0138,0180,0950,0000,10F4  ;3668  Z?,D_RC[0A]
U 10F4, 0000,003D,0180,0800,0000,115C  ;3669 =0 ERROR1     ;DIDN'T SEE READ DATA AND HOLD
          ;3670    ;================================================================
          ;3671    ;SEE READ DATA
          ;3672    ;================================================================
U 10F5, 0819,0014,0580,09D0,0000,12EC  ;3673 S05150: RC[0A]_ALU,D_D+K[.1]  ;SEE READ DATA
U 12EC, 0800,003C,0180,0A30,0000,12ED  ;3674  D_R[6]
U 12ED, 0001,003C,61F0,2DE0,0000,12EE  ;3675  RC[0C]_D,Q_ID[SILO]
U 12EE, 0001,203C,0180,09D8,0000,12EF  ;3676  RC[0B]_Q
U 12EF, 001D,0020,0180,0800,0010,12F0  ;3677  ALU_D[XOR]Q,CLK.UBCC
U 12F0, 0810,0138,0180,0950,0000,10F6  ;3678  Z?,D_RC[0A]
U 10F6, 0000,003D,0180,0800,0000,115C  ;3679 =0 ERROR1     ;DIDN'T SEE READ DATA
          ;3680    ;================================================================
          ;3681    ;SEE ACK (OR NULL<I/O REFERENCE>) FROM CPU
          ;3682    ;================================================================
U 10F7, 0819,0014,0580,09D0,0000,12F1  ;3683  RC[0A]_ALU,D_D+K[.1]  ;SEE ACK FROM CPU OR NULL(I/O)
```

```
U 12F1, 0010.0038.0180.0940.0010.12F2  ;3684  ALU_RC[8],CLK.UBCC  ;TEST I/O FLAG
U 12F2, 0800.013C.0180.0A68.0000.10F8  ;3685  Z?,D_R[0D]   ;NON I/O REFERENCE
U 10F8, 0F00.003C.0180.0800.0000.10F9  ;3686 =0 D_0    ;I/O REFERENCE
U 10F9, 0001.003C.61F0.2DE0.0000.12F3  ;3687  RC[0C]_D,Q_ID[SILO]
U 12F3, 0001.203C.0180.09D8.0000.12F4  ;3688  RC[0B]_Q
U 12F4, 001D.0020.0180.0800.0010.12F5  ;3689  ALU_D[XOR]Q,CLK.UBCC
U 12F5, 0810.0138.0180.0950.0000.10FA  ;3690  Z?,D_RC[0A]
U 10FA, 0000.003D.0180.0800.0000.115C  ;3691 =0 ERROR1    ;DIDN'T SEE ACK(NULL) FROM CPU
            ;3692   ;=============================================================
            ;3693   ;SEE 2ND ACK FROM MEMORY
            ;3694   ;  (1ST ACK IF I/O REFERENCE)
            ;3695   ;=============================================================
U 10FB, 0819.0014.0580.09D0.0000.12F6  ;3696  RC[0A]_ALU,D_D+K[.1]
U 12F6, 0800.003C.61F0.2E68.0000.12F7  ;3697  Q_ID[SILO],D_R[0D]
U 12F7, 0001.003C.0180.09E0.0000.12F8  ;3698  RC[0C]_D   ;SET EXPECTED DATA
U 12F8, 0001.203C.0180.09D8.0000.12F9  ;3699  RC[0B]_Q
U 12F9, 001D.2020.0180.0800.0010.12FA  ;3700  ALU_Q[XOR]D,CLK.UBCC
U 12FA, 0810.0138.0180.0950.0000.10FC  ;3701  Z?,D_RC[0A]
U 10FC, 0000.003D.0180.0800.0000.115C  ;3702 =0 ERROR1    ;DIDN'T SEE 2ND ACK IN SILO
            ;3703   ;=============================================================
            ;3704   ;SEE NULL CYCLES UNTILL COUNT = F
            ;3705   ;=============================================================
U 10FD, 0819.0014.0580.09D0.0000.12FB  ;3706 S05158: RC[0A]_ALU,D_D+K[.1]  ;SEE NULL CYCLES IN
U 12FB, 0003.003C.61F0.2DE0.0000.12FC  ;3707  RC[0C]_0,Q_ID[SILO]  ; SILO (UNTIL COUNT=F)
U 12FC, 0001.203C.0180.09D8.0010.12FD  ;3708  RC[0B]_Q,CLK.UBCC
U 12FD, 0810.0138.0180.0950.0000.10FE  ;3709  Z?,D_RC[0A]
U 10FE, 0000.003D.0180.0800.0000.115C  ;3710 =0 ERROR1    ;DIDN'T SEE A NULL CYCLE
U 10FF, 0019.0020.6180.0800.0010.12FE  ;3711  ALU_D[XOR]K[.F],CLK.UBCC ;COUNT=F?
U 12FE, 0810.0138.0180.0950.0000.110A  ;3712  Z?,D_RC[0A]
U 110A, 0000.003C.0180.0800.0000.10FD  ;3713 =0 J/S05158   ;NO,REPEAT LOOP
            ;3714   ;=============================================================
            ;3715   ;CHECK OPERATION POINTER AND ADDRESS FLAG
            ;3716   ;=============================================================
U 110B, 0810.0038.0180.0948.0000.12FF  ;3717  D_RC[9]   ;ALL OPERATION DONE
U 12FF, 0019.0020.0D80.0800.0010.1300  ;3718  ALU_D[XOR]K[.3],CLK.UBCC
U 1300, 0000.013C.0180.0800.0000.1110  ;3719  Z?
U 1110, 0819.0014.0580.09C8.0000.10D2  ;3720 =0 RC[9]_ALU,D_D+K[.1],J/S05105 ;GO REPEAT LOOP WITH
            ;3721       ; NOW MCT FUNCTION
U 1111, 0010.0038.0180.0940.0010.1301  ;3722  ALU_RC[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 1301, 0000.013C.0180.0800.0000.1112  ;3723  Z?
U 1112, 0000.003C.0180.0800.0000.1302  ;3724 =0 J/S05160   ;END TEST
U 1113, 0018.0038.0580.09C0.0000.12BB  ;3725  RC[8]_K[.1],J/S05100
U 1302, 0000.003C.0180.0800.0000.1114  ;3726 S05160: NOP
            ;3727 ;
            ;3728 ;=============================================================================
            ;3729 ;*
            ;3730 ;SUBTEST 3
            ;3731 ;  MASKED WRITES, ADDR BIT #29=0,1
            ;3732 ;
            ;3733 ;  FUNCTION OPERATION POINTER
            ;3734 ;  -------- ------------------------------------
            ;3735 ;  MCT/WRITE.P  (0)
            ;3736 ;  MCT/WRITE.V.NOCHK (1)
            ;3737 ;  MCT/WRITE.V.WCHK (2)
            ;3738 ;
```

```
                ;3739 ;  SEE SILO INTERRUPT AND CHECK CONTENTS OF SILO
                ;3740 ;
                ;3741 ;  SEE THE FOLLOWING:
                ;3742 ;
                ;3743 ;  NULL (4 TICKS)
                ;3744 ;  CPU ARB (IF I.D. NOT=10)
                ;3745 ;  C/A (MASKED WRITES), AND HOLD
                ;3746 ;  WRITE DATA
                ;3747 ;  ACK
                ;3748 ;  ACK
                ;3749 ;  NULL (UNTILL COUNT = F)
                ;3750 ;
                ;3751 ; ERROR LOGOUT
                ;3752 ;
                ;3753 ; DATA: EXPECTED
                ;3754 ;  DATA FROM SILO
                ;3755 ;  SILO COUNT, 0=1ST LOCATION AFTER LOCK
                ;3756 ;  OPERATION POINTER (0-2)
                ;3757 ;  ADDRESS FLAG (0=NON I/O SPACE, 1=I/O SPACE)
                ;3758 ;  TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
                ;3759 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
                ;3760 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
                ;3761 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;3762 ;  INTERRUPT FLAG:
                ;3763 ;   0=>NO INTERRUPT OCCURRED
                ;3764 ;   1=>AN INTERRUPT OCCURRED
                ;3765 ;-
                ;3766 ;================================================================================
                ;3767 ;
U 1114, 0000.003D.0180.0800.0000.1160  ;3768 =0 SUBTEST
U 1115, 0003.003C.0180.09C0.0000.1303  ;3769  RC[8]_0    ;SET ADDRESS FLAG
U 1303, 0003.003C.0180.09C8.0000.1116  ;3770 S05200: RC[9]_0    ;SET INITIAL OPERATION
U 1116, 0000.003D.0180.0800.0000.115B  ;3771 =0 ERLOOP     ; POINTER
U 1117, 0000.003C.0180.0800.0000.1118  ;3772 RETRY2: NOP
U 1118, 0000.003D.0180.0800.0000.11F5  ;3773 =0 CALL[INIT]
U 1119, 0010.0038.0180.0970.0200.1304  ;3774  VA_RC[0E]    ; Load VA with CNFG Reg. A address
U 1304, 0000.003C.0180.C800.0000.1305  ;3775  D[LONG]_CACHE.P    ; Read it to clear any Fault bits
U 1305, 0000.003C.0180.0800.0000.1306  ;3776  NOP
U 1306, 0F10.0038.0180.0968.0200.1307  ;3777  VA_RC[0D],D_0    ;DO I/O WRITE TO
U 1307, 0000.003C.0180.A800.0000.111A  ;3778  CACHE.P_D[LONG]    ; CLEAR IMPLICT FLOP        *****
                ;3779 =0 D_K[.4],
U 111A, 0818.0039.1180.0800.0000.1217  ;3780   CALL[WAIT.LOOP]
U 111B, 0000.003C.0180.0800.0000.1030  ;3781  NOP
U 1030, 0000.003D.0180.0800.0000.1042  ;3782 =00 CALL[NOINTR]
U 1031, 0000.003D.0180.0800.0000.115C  ;3783  ERROR1     ;UNEXPECTED INTERRUPT
U 1032, 0010.0038.0180.0940.0010.1033  ;3784 S05205: ALU_RC[8],CLK.UBCC  ; SET ADDRESS FOR MCT/
U 1033, 0018.0138.1980.0800.0200.111C  ;3785  Z?,VA_K[ZERO]    ; FUNCTION
U 111C, 0010.0038.0180.0968.0200.111D  ;3786 =0 VA_RC[0D]    ;I/O REFERENCE
U 111D, 0800.003C.0180.0A58.0000.1308  ;3787  D_R[0B]
U 1308, 0810.0038.7180.3D48.0000.1309  ;3788  ID[COMP]_D,D_RC[9]  ;ENABLE SILO, SET OPERATION
U 1309, 0000.193C.0180.0800.0000.102C  ;3789  D1-0?     ;BRANCH TO OPERATION
U 102C, 0000.003C.0180.A800.0000.111E  ;3790 =1100 MCT/WRITE.P,J/S05210  ; (0)
U 102D, 0000.003C.0180.2800.0000.111E  ;3791  MCT/WRITE.V.NOCHK,J/S05210 ; (1)
U 102E, 0000.003C.0180.3000.0000.111E  ;3792  MCT/WRITE.V.WCHK,J/S05210 ; (2)
U 102F, 0000.003D.0180.0800.0000.115C  ;3793  ERROR1     ;SHOULD NEVER GET HERE
```

```
                ;3794 =0
                ;3795 S05210: D_K[.14],
U 111E. 0818.0039,2180.0800,0000,1217  ;3796   CALL[WAIT.LOOP]  ;LET SILO FILL
U 111F. 0000.003C,0180.0800,0000,1034  ;3797  NOP
U 1034. 0000.003D,0180.0800,0000,11FE  ;3798 =00 CALL[SILINT]   ;SEE SILO INTERRUPT
U 1035. 0000.003D,0180.0800,0000,115C  ;3799  ERROR1     ;GOT WRONG INTERRUPT
                ;3800   ;================================================================
                ;3801   ;
                ;3802   ;CHECK CONTENTS OF SILO
                ;3803   ;
                ;3804   ;================================================================
U 1036. 0003.003C,0180,09D0,0000,1037  ;3805  RC[0A]_0    ;RESET COUNTER
U 1037. 0003.003C,0180,09E0,0000,130A  ;3806  RC[0C]_0    ;SET EXPECTED DATA
                ;3807   ;================================================================
                ;3808   ;SEE 4 NULL CYCLE IN SILO
                ;3809   ;================================================================
U 130A. 0000.003C,61F0,2C00,0000,130B  ;3810 S05220: Q_ID[SILO]
U 130B. 0001.203C,0180,09D8,0010,130C  ;3811  RC[0B]_Q,CLK.UBCC
U 130C. 0810.0138,0180,0950,0000,1120  ;3812  Z?,D_RC[0A]
U 1120. 0000.003D,0180,0800,0000,115C  ;3813 =0 ERROR1     ;DIDN'T SEE NULL CYCLE
U 1121. 0019.0020,0D80,0800,0010,130D  ;3814  ALU_D[XOR]K[.3],CLK.UBCC ;4 CYCLES YET
U 130D. 0819.0114,0580,0800,0000,1122  ;3815  Z?,D_D+K[.1]
U 1122. 0001.003C,0180,09D0,0000,130A  ;3816 =0 RC[0A]_D,J/S05220  ;REPEAT LOOP
U 1123. 0818.0038,6580,0800,0000,130E  ;3817  D_K[.10]
U 130E. 0011.0020,0180,0910,0010,130F  ;3818  ALU_D[XOR]RC[2],CLK.UBCC ;I.D.=10?
U 130F. 0810.0138,0180,0950,0000,1124  ;3819  Z?,D_RC[0A]
U 1124. 0000.003C,0180,0800,0000,1310  ;3820 =0 J/S05230    ;ID NOT 10, SEE ARB CYCLE
U 1125. 0000.003C,0180,0800,0000,1127  ;3821  J/S05240    ;ID=10,SEE C/A CYCLE
                ;3822   ;================================================================
                ;3823   ;SEE CPU ARB CYCLE
                ;3824   ;================================================================
U 1310. 0819.0014,0580,09D0,0000,1311  ;3825 S05230: RC[0A]_ALU,D_D+K[.1]
U 1311. 0800.003C,61F0,2E50,0000,1312  ;3826  D_R[0A],Q_ID[SILO]
U 1312. 0001.003C,0180,09E0,0000,1313  ;3827  RC[0C]_D
U 1313. 0001.203C,0180,09D8,0000,1314  ;3828  RC[0B]_Q
U 1314. 001D.0020,0180,0800,0010,1315  ;3829  ALU_D[XOR]Q,CLK.UBCC
U 1315. 0810.0138,0180,0960,0000,1126  ;3830  Z?,D_RC[0C]
U 1126. 0000.003D,0180,0800,0000,115C  ;3831 =0 ERROR1     ;DIDN'T SEE CPU ARB
                ;3832   ;================================================================
                ;3833   ;SEE MASKED WRITE C/A AND HOLD
                ;3834   ;================================================================
U 1127. 0819.0014,0580,09D0,0000,1316  ;3835 S05240: RC[0A]_ALU,D_D+K[.1]
U 1316. 0818.0030,0580,0A20,0000,1317  ;3836  D_ALU,ALU_R[4].OR.K[.1]
U 1317. 0001.003C,61F0,2DE0,0000,1318  ;3837  RC[0C]_D,Q_ID[SILO]
U 1318. 0001.203C,0180,09D8,0000,1319  ;3838  RC[0B]_Q
U 1319. 001D.0020,0180,0800,0010,131A  ;3839  ALU_D[XOR]Q,CLK.UBCC
U 131A. 0810.0138,0180,0950,0000,1128  ;3840  Z?,D_RC[0A]
U 1128. 0000.003D,0180,0800,0000,115C  ;3841 =0 ERROR1     ;DIDN'T SEE C/A AND HOLD
                ;3842   ;================================================================
                ;3843   ;SEE WRITE DATA
                ;3844   ;================================================================
U 1129. 0819.0014,0580,09D0,0000,131B  ;3845  RC[0A]_ALU,D_D+K[.1]
U 131B. 0800.003C,61F0,2E40,0000,131C  ;3846  D_R[8],Q_ID[SILO]  ;SEE ALL ONES FOR MASKED BITS
U 131C. 0001.003C,0180,09E0,0000,131D  ;3847  RC[0C]_D   ; EVEN THOUGH THEY ARE NOT USED
U 131D. 0001.203C,0180,09D8,0000,131E  ;3848  RC[0B]_Q
```

```
U 131E. 001D.0020.0180.0800.0010.131F  ;3849  ALU_D[XOR]Q,CLK.UBCC
U 131F. 0810.0138.1980.0950.0084.712A  ;3850  Z?,D_RC[0A],SC_K[ZERO]
U 112A. 0000.003D.0180.0800.0000.115C  ;3851 =0 ERROR1     ;DIDN'T SEE WRITE DATA
          ;3852       ; IN SILO
          ;3853    ;================================================================
          ;3854    ;SEE 2 ACKS FROM MEMORY
          ;3855    ;================================================================
U 112B. 0819.0014.0580.09D0.0084.9320  ;3856 S05250: RC[0A]_ALU,D_D+K[.1],SC_SC+K[.1]
U 1320. 0800.003C.61F0.2E68.0000.1321  ;3857  D_R[0D],Q_ID[SILO]
U 1321. 0001.003C.0180.09E0.0000.1322  ;3858  RC[0C]_D
U 1322. 0001.203C.0180.09D8.0000.1323  ;3859  RC[0B]_Q
U 1323. 001D.0020.0180.0800.0010.1324  ;3860  ALU_D[XOR]Q,CLK.UBCC
U 1324. 0810.0138.0180.0950.0000.112C  ;3861  Z?,D_RC[0A]
U 112C. 0000.003D.0180.0800.0000.115C  ;3862 =0 ERROR1     ;DIDN'T GET ACK FROM
          ;3863       ; MEMORY
U 112D. 0018.0038.1DC0.0800.0000.1325  ;3864  Q_ALU,ALU_SC    ;SC=Z?
U 1325. 0019.2020.0980.0800.0010.1326  ;3865  ALU_Q.XOR.K[.2],CLK.UBCC
U 1326. 0000.013C.0180.0800.0000.112E  ;3866  Z?
U 112E. 0000.003C.0180.0800.0000.112B  ;3867 =0 J/S05250    ;GO REPEAT LOOP
          ;3868    ;================================================================
          ;3869    ;SEE NULL CYCLE UNTIL COUNT=F (RC C)
          ;3870    ;================================================================
U 112F. 0819.0014.0580.09D0.0000.1327  ;3871 S05260: RC[0A]_ALU,D_D+K[.1]
U 1327. 0003.003C.61F0.2DE0.0000.1328  ;3872  RC[0C]_0,Q_ID[SILO]
U 1328. 0001.203C.0180.09D8.0010.1329  ;3873  RC[0B]_Q,CLK.UBCC
U 1329. 0810.0138.0180.0950.0000.1130  ;3874  Z?,D_RC[0A]
U 1130. 0000.003D.0180.0800.0000.115C  ;3875 =0 ERROR1     ;DIDN'T SEE NULL CYCLE
U 1131. 0019.0020.6180.0800.0010.132A  ;3876  ALU_D[XOR]K[.F],CLK.UBCC
U 132A. 0810.0138.0180.0950.0000.1132  ;3877  Z?,D_RC[0A]
U 1132. 0000.003C.0180.0800.0000.112F  ;3878 =0 J/S05260    ;GO REPEAT LOOP
          ;3879    ;================================================================
          ;3880    ; CHECK OPERATION POINTER AND ADDRESS FLAG
          ;3881    ;================================================================
U 1133. 0810.0038.0180.0948.0000.132B  ;3882  D_RC[9]
U 132B. 0019.0020.0980.0800.0010.132C  ;3883  ALU_D[XOR]K[.2],CLK.UBCC ;ALL OPERATIONS DONE?
U 132C. 0000.013C.0180.0800.0000.1134  ;3884  Z?
U 1134. 0819.0014.0580.09C8.0000.1032  ;3885 =0 RC[9]_ALU,D_D+K[.1],J/S05205 ;REPEAT LOOP,NEW OPERATION
U 1135. 0010.0038.0180.0940.0010.132D  ;3886  ALU_RC[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 132D. 0000.013C.0180.0800.0000.1136  ;3887  Z?
U 1136. 0000.003C.0180.0800.0000.132E  ;3888 =0 J/S05270    ;FLAG SET
U 1137. 0018.0038.0580.09C0.0000.1303  ;3889  RC[8]_K[.1],J/S05200  ;REPEAT,WITH FLAG SET
U 132E. 0000.003C.0180.0800.0000.1138  ;3890 S05270: NOP     ;END
          ;3891
```

```
            ;3892  .PAGE "TEST 19B RESERVED"
            ;3893 =0
U 1138, 0000.003D.0180.0800.0000.1145  ;3894 T19B: NEWTST
U 1139, 0000.003C.0180.0800.0000.113A  ;3895  NOP
            ;3896
```

```
            ;3897  .PAGE "TEST 19C RESERVED"
            ;3898 =0
U 113A, 0000,003D,0180,0800,0000,1145  ;3899 T19C: NEWTST
U 113B, 0000,003C,0180,0800,0000,113C  ;3900  NOP
            ;3901
```

```
          ;3902  .PAGE "DUMMY NEWTST"
          ;3903 =0
U 113C, 0000,003D,0180,0800,0000,1145  ;3904 DUMMY: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0510
ESKAR42.MCR                    MICRO2  1P(00)    29-JUL-85  15:00:39     Page  12
:                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1911: 1878 1885= 1901= 2633= 2634= 2635= 2637=
U 1008  2356= 2357= 2358= 2359= 2360= 2361= 2362= 2363=
U 1010  2681= 2682= 2683= 2685= 3327= 3328= 3329= 1879
U 1018  3342= 3343= 3344= 3347= 3561= 3562= 3563= 3564=
U 1020  3348= 3353= 3367= 3370= 3421= 3422= 3423= 3424=
U 1028  3569= 3570= 3576= 3577= 3790= 3791= 3792= 3793=
U 1030  3782= 3783= 3784= 3785= 3798= 3799= 3805= 3806=
U 1038  1902= 1903= 1955= 1957= 1968= 1969= 1998= 1999=
U 1040  2025= 2026= 2131= 2132= 2135= 2136= 2138= 2139=
U 1048  2143= 2144= 2216= 2217= 2291= 2292= 1979: 1880
U 1050  2335= 2336= 2344= 2345= 2346= 2348= 2351= 2352=
U 1058  2393= 2394= 2007: 1881 2395= 2396= 2034: 1882
U 1060  2403= 2404= 2410= 2412= 2415= 2416= 2435= 2436=
U 1068  2438= 2439= 2531= 2532= 2535= 2536= 2547= 2548=
U 1070  2551= 2552= 2588= 2589= 2702= 2703= 2705= 2706=
U 1078  2708= 2709= 2726= 2727= 2728= 2729= 2731= 2732=
U 1080  2733= 2734= 2735= 2736= 2754= 2755= 2775= 2776=
U 1088  2778= 2779= 2785= 2786= 2829= 2830= 2831= 2833=
U 1090  2834= 2835= 2836= 2837= 2838= 2839= 2840= 2841=
U 1098  2842= 2843= 2845= 2846= 2882= 2883= 2894= 2895=
U 10A0  2994= 2995= 3072= 3073= 3075= 3076= 3110= 3111=
U 10A8  3122= 3123= 3130= 3131= 3132= 3133= 3141= 3142=
U 10B0  3143= 3144= 3148= 3149= 3151= 3152= 3156= 3157=
U 10B8  3168= 3169= 3175= 3176= 3183= 3184= 3324= 3326=
U 10C0  3382= 3383= 3393= 3394= 3395= 3396= 3405= 3406=
U 10C8  3418= 3419= 3434= 3435= 3454= 3455= 3544= 3545=
U 10D0  3547= 3548= 3551= 3552= 3554= 3555= 3557= 3558=
U 10D8  3567= 3568= 3582= 3583= 3585= 3586= 3589= 3590=
U 10E0  3600= 3604= 3607= 3608= 3612= 3613= 3617= 3621=
U 10E8  3625= 3626= 3628= 3629= 3632= 3637= 3644= 3649=
U 10F0  3656= 3657= 3659= 3663= 3669= 3673= 3679= 3683=
U 10F8  3686= 3687= 3691= 3696= 3702= 3706= 3710= 3711=
U 1100  1865: 1866: 1867: 1868: 1869: 1870: 1871: 1872:
U 1108  1873: 1883 3713= 3717= 1874: 1875: 1876: 1877:
U 1110  3720= 3722= 3724= 3725= 3768= 3769= 3771= 3772=
U 1118  3773= 3774= 3780= 3781= 3786= 3787= 3796= 3797=
U 1120  3813= 3814= 3816= 3817= 3820= 3821= 3831= 3835=
U 1128  3841= 3845= 3851= 3856= 3862= 3864= 3867= 3871=
U 1130  3875= 3876= 3878= 3882= 3885= 3886= 3888= 3889=
U 1138  3894= 3895= 3899= 3900= 3904= 1884 1904 1905
U 1140  1906 1907 1908 1909 1910 1944 1945 1946
U 1148  1947 1948 1949 1950 1951 1952 1953 1954
U 1150  1958 1959 1960 1961 1962 2057: 1963 1964
U 1158  1965 1966 1967 1973 1977 1978 2005 2006
U 1160  2068 2069 2079 2080 2089 2090 2091 2104
U 1168  2105 2106 2107 2108 2133 2134 2140 2141
U 1170  2142 2145 2146 2147 2148 2149 2150 2151
U 1178  2152 2172 2174 2175 2176 2177 2178 2183
U 1180  2209 2210 2211 2212 2214 2215 2244 2245
U 1188  2246 2265 2266 2267 2268 2293 2294 2295
U 1190  2296 2337 2338 2339 2340 2349 2350 2354
U 1198  2355 2368 2369 2371 2372 2373 2375 2380
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2381 2382 2384 2389 2391 2392 2400 2401
U 11A8  2405 2406 2407 2408 2409 2417 2418 2419
U 11B0  2421 2422 2423 2424 2425 2426 2431 2433
U 11B8  2434 2437 2462 2463 2464 2465 2495 2496
U 11C0  2498 2499 2500 2527 2529 2530 2534 2538
U 11C8  2539 2540 2541 2543 2544 2545 2546 2549
U 11D0  2550 2577 2578 2579 2580 2581 2582 2583
U 11D8  2584 2586 2587 2700 2737 2738 2739 2749
U 11E0  2750 2751 2752 2753 2777 2780 2781 2782
U 11E8  2783 2784 2787 2794 2795 2796 2810 2811
U 11F0  2812 2813 2814 2815 2816 2828 2847 2848
U 11F8  2876 2877 2878 2879 2880 2881 2888 2889
U 1200  2890 2891 2892 2893 2907 2908 2909 2920
U 1208  2921 2922 2923 2934 2935 2936 2937 2948
U 1210  2949 2950 2951 2961 2962 2963 2964 2989
U 1218  3017 3059 3061 3062 3063 3064 3065 3068
U 1220  3070 3071 3112 3113 3114 3115 3116 3117
U 1228  3118 3119 3120 3121 3124 3125 3126 3127
U 1230  3128 3134 3135 3136 3137 3138 3139 3140
U 1238  3145 3146 3147 3153 3154 3155 3158 3159
U 1240  3160 3161 3162 3163 3164 3165 3166 3167
U 1248  3170 3171 3172 3173 3174 3177 3178 3179
U 1250  3180 3181 3182 3185 3186 3187 3332 3333
U 1258  3334 3336 3338 3339 3340 3341 3384 3385
U 1260  3386 3387 3388 3392 3397 3398 3399 3400
U 1268  3401 3402 3403 3404 3408 3409 3410 3411
U 1270  3412 3413 3414 3415 3420 3425 3426 3427
U 1278  3428 3429 3430 3431 3432 3433 3436 3437
U 1280  3438 3439 3440 3441 3442 3443 3444 3445
U 1288  3446 3447 3448 3449 3450 3451 3452 3453
U 1290  3456 3457 3458 3459 3460 3461 3462 3463
U 1298  3464 3465 3466 3467 3468 3469 3470 3471
U 12A0  3472 3473 3474 3475 3476 3477 3478 3479
U 12A8  3480 3481 2058: 3482 3483 3484 3485 3486
U 12B0  3487 3488 3489 3490 3491 3492 3493 3494
U 12B8  3495 3496 3497 3546 3553 3556 3559 3560
U 12C0  3579 3580 3581 3584 3587 3588 3594 3595
U 12C8  3596 3597 3598 3599 3605 3606 3609 3610
U 12D0  3611 3614 3615 3616 3622 3623 3624 3627
U 12D8  3630 3631 3638 3639 3640 3641 3642 3643
U 12E0  3650 3651 3652 3653 3654 3655 3658 3664
U 12E8  3665 3666 3667 3668 3674 3675 3676 3677
U 12F0  3678 3684 3685 3688 3689 3690 3697 3698
U 12F8  3699 3700 3701 3707 3708 3709 3712 3718
U 1300  3719 3725 3726 3770 3775 3776 3777 3778
U 1308  3788 3789 3810 3811 3812 3815 3818 3819
U 1310  3825 3826 3827 3828 3829 3830 3836 3837
U 1318  3838 3839 3840 3846 3847 3848 3849 3850
U 1320  3857 3858 3859 3860 3861 3865 3866 3872
U 1328  3873 3874 3877 3883 3884 3887 3890
U 1330 - 13EF Unused
U 13F0  2041: 2042: 2043: 2044: 2045: 2046: 2047: 2048:
U 13F8  2049: 2050: 2051: 2052: 2053: 2054: 2055: 2056:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR42	831
Used	831
Remaining	

Total microwords used in memory U: 831
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR42.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1848 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1915 Micro Monitor Interface Routines
; 1916   Newtest Routine
; 1973   ERLOOP ROUTINE
; 1977   ERROR1 ROUTINE
; 1983   ERROR1 WITH PARAMETER
; 2005   ERROR2 ROUTINE
; 2011   ERROR 2 WITH PARAMETER
; 2032   CALLCON ROUTINE
; 2038   SCRATCH PAD LOCATIONS
; 2062   SUBTEST ROUTINE
; 2073   SUBTEST 1 ROUTINE
; 2084   SETRCF ROUTINE
; 2095   ENBLFLT ROUTINE
; 2112   NOINTR ROUTINE
; 2157   GENREATE CONSTANT ROUTINE (DCE)
; 2183   DELAY ROUTINE
; 2187 DIAGNOSTIC SPECIFIC ROUTINES
; 2189   CHECK UTRAP
; 2221   Set Trap Mask
; 2250   Disable Cache Routine
; 2272   Initialize the SBI
; 2301   FIND MS780-E MEMORY
; 2443   Generate IO Address
; 2470   Set Starting Address
; 2504   ENABLE NEXT MS780-E
; 2555   Select Controller
; 2593   SELECT LOWER CONTROLLER
; 2641   SELECT UPPER CONTROLLER
; 2690   CLRSBI ROUTINE
; 2713   SLOLCK ROUTINE
; 2745   CLCPTO ROUTINE
; 2754   INIT ROUTINE
; 2786   CHECK FOR INTERRUPT VECTORS
; 2833   SETPSL ROUTINE
; 2847   CLRTIM ROUTINE
; 2861   CLRFLT ROUTINE
; 2875   CLRECC ROUTINE
; 2889   ENBECC ROUTINE
; 2902   Wait Loop Routine
; 2933   RESET SUBTEST NUMBER
; 2955   WAIT REFRESH SUBROUTINE
; 3017 TEST 19D CHECK SBI COMMANDS AND STALL USING SILO
; 3772 TEST 19E RESERVED
; 3777 TEST 19F RESERVED
; 3782 TEST 1A0 RESERVED
; 3787 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR43
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ; 2.1 Joe Zagarella  11-jul-1985
;1836  ;
;1837  ;  With only the upper controller configured on
;1838  ;  an ms780-e/h, test 19d would fail because it
;1839  ;  only checked for the lower controller. Added
;1840  ;  code at label "try.upper" to check for the upper
;1841  ;  controller.
;1842  ;
;1843  ;
;1844  ;
;1845  ;*****************************************************************************
;1846
```

```
                ;1847  .PAGE
                ;1848  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
                ;1849 ;*
                ;1850 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1851 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
                ;1852 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1853 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1854 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1855 ;
                ;1856 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1857 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1858 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1859 ; R[0F] AND DO A RETURN0.
                ;1860 ;
                ;1861 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1862 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1863 ; TO THE CONSOLE.
                ;1864 ;
                ;1865 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1866 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7001  ;1867 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7001  ;1868 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7001  ;1869 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7001  ;1870 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7001  ;1871 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7001  ;1872 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7001  ;1873 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7001  ;1874 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7001  ;1875 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D, 0000.003C.8980.0800.0084.7001  ;1876 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7001  ;1877 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1878 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.1017  ;1879 TRPH0: Q_ID[USTACK]
U 1017, 0C00.003C.01E0.0800.0000.101B  ;1880  Q_D,D_Q
U 101B, 0000.003C.8180.3C00.0000.1023  ;1881  ID[USTACK]_D
U 1023, 0C00.003C.01E0.0800.0000.102B  ;1882  Q_D,D_Q
U 102B, 0000.003C.01C0.0A78.0000.104F  ;1883  Q_R[0F]
U 104F, 0001.2024.0180.0800.0010.105B  ;1884  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 105B, 0000.013C.0180.0800.0000.1002  ;1885  Z?    ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1886 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1887 ;*
                ;1888 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1889 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1890 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1891 ;
                ;1892 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1893 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1894 ;       FAULT STATUS REGISTER
                ;1895 ;       SBI ERROR REGISTER
                ;1896 ;       TIMEOUT ADDRESS REGISTER
                ;1897 ;       MAINTENANCE REGISTER
                ;1898 ;       CACHE PARITY REGISTER
                ;1899 ;       TB ERROR REGISTER 1
                ;1900 ;       TB ERROR REGISTER 0
                ;1901 ;-
```

```
U 1003, 0018,0038,1D80,09E8,0000,101C  ;1902 TRPH1: RC[0D]_SC
U 101C, 0000,003D,D980,0800,0084,7145  ;1903 =0 SETRCF[.9]
U 101D, 0000,003C,49F0,2C00,0000,105F  ;1904  Q_ID[TBER0]
U 105F, 0001,203C,4DF0,2DB0,0000,1109  ;1905  RC[6]_Q,Q_ID[TBER1]
U 1109, 0001,203C,65F0,2DB8,0000,1122  ;1906  RC[7]_Q,Q_ID[SBI.ERR]
U 1122, 0001,203C,6DF0,2DD8,0000,1123  ;1907  RC[0B]_Q,Q_ID[FAULT]
U 1123, 0001,203C,75F0,2DE0,0000,1124  ;1908  RC[0C]_Q,Q_ID[MAINT]
U 1124, 0001,203C,79F0,2DC8,0000,1125  ;1909  RC[9]_Q,Q_ID[PARITY]
U 1125, 0001,203C,69F0,2DC0,0000,1126  ;1910  RC[8]_Q,Q_ID[TIME.ADDR]
U 1126, 0001,203C,81F0,2DD0,0000,1000  ;1911  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001,203D,0180,09F0,0000,113D  ;1912 1000: RC[0E]_Q,ERROR1
            ;1913
```

```
                ;1914  .PAGE
                ;1915  .TOC " Micro Monitor Interface Routines"
                ;1916  .TOC "   Newtest Routine"
                ;1917 ;++
                ;1918 ; FUNCTIONAL DESCRIPTION:
                ;1919 ;
                ;1920 ; This routine Initializes the following registers:
                ;1921 ;  PSL    to 1F
                ;1922 ;  CES    to 0
                ;1923 ;  ACC.CS    to disable accellerator
                ;1924 ;  SIR    to 0
                ;1925 ;  CLK.CS    to stop interval timer
                ;1926 ;  TBER0    to disable the TB
                ;1927 ;  SBI.MAINT to     0
                ;1928 ;  SBI.ERR    to 0
                ;1929 ;  FAULT    to 0
                ;1930 ;  COMP    to 0
                ;1931 ;  RC[0E]    to  0
                ;1932 ;  R[0F]    to 0
                ;1933 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1934 ; A SCOPE call is then made to the Monitor.
                ;1935 ;
                ;1936 ; CALLING CONSTRAINT:
                ;1937 ;
                ;1938 ; =0
                ;1939 ;
                ;1940 ; SIDE EFFECTS:
                ;1941 ;
                ;1942 ; D Reg is destroyed
                ;1943 ; SC is destroyed
                ;1944 ;--
U 1127, 0000.003C.65F8,0800.0084.7128   ;1945 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1128, 0818.0038.8D80,0800.0000.1129   ;1946  d_k[.1f]  ; D=1F
U 1129, 0D00.003C.0180,0800.0000.112A   ;1947  d_dal.sc  ; D=001F0000
U 112A, 0000.003C.3D80,3C00.0000.112B   ;1948  id[psl]_d  ; psl = 001F0000
U 112B, 0818.0038.0580,0800.0000.112C   ;1949  d_k[.1]   ; Clear the CES register
U 112C, 0D00.003C.0180,0800.0000.112D   ;1950  d_dal.sc  ; D = 00010000
U 112D, 0F00.003C.3180,3C00.0000.112E   ;1951  id[ces]_d,d_0  ; ces = 00010000
U 112E, 0000.003C.5D80,3C00.0000.112F   ;1952  id[acc.cs]_d  ; acc.cs = 0
U 112F, 0000.003C.3980,3C00.0000.1130   ;1953  id[sir]_d  ; sir = 0
U 1130, 0000.003C.2980,3C00.0000.1131   ;1954  id[clk.cs]_d  ; clk.cs = 0
U 1131, 0000.003C.4980,3C00.0000.1034   ;1955  id[tber0]_d  ; tber0 = 0
U 1034, 0000.003D.0180,0800.0000.11BD   ;1956 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1957  intrpt.strobe,  ; Strobe interrupts
U 1035, 0818.0038.C180,0800.4000.1132   ;1958       d_k[.ffff]  ; Setup to clear SBI error register and
U 1132, 0801.0028.6580,3C00.0000.1133   ;1959  id[sbi.err]_d,d_not.d ; and fault register
U 1133, 0F00.003C.6D80,3C00.0000.1134   ;1960  id[fault]_d,d_0  ; ...
U 1134, 0000.003C.6D80,3C00.0000.1135   ;1961  id[fault]_d  ; fault = 0
U 1135, 0080.003C.7580,3C00.0000.1136   ;1962  id[maint]_d  ; MAINT = 0
U 1136, 0000.003C.6580,3C00.0000.1137   ;1963  id[sbi.err]_d  ; sbi error reg = 0
U 1137, 0000.003C.7180,3C00.0000.1138   ;1964  id[comp]_d  ; comp reg = 0
U 1138, 0000.003C.E580,3C00.0000.1139   ;1965  id[39]_d  ; Clear code for subroutine START.TEST
U 1139, 0001.003C.0180,09F0.0000.113A   ;1966  rc[0e]_d  ;
U 113A, 0001.003C.0180,0AF8.0000.113B   ;1967  r[0f]_d  ; Clear expected trap mask
U 113B, 0001.003C.0180,09F8.0000.1036   ;1968  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1036, 0000,003D,0180,0800,0000,116B  ;1969 =0 call[disable.cache] ; Disable the cache
U 1037, 0F03,003C,0180,09E8,0000,105E  ;1970  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
            ;1971
```

```
            ;1972  .PAGE
            ;1973  .TOC "    ERLOOP ROUTINE"
U 113C, 0818,0038,0980,0800,0000,105E  ;1974 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
            ;1975
```

```
            ;1976   .PAGE
            ;1977   .TOC "    ERROR1 ROUTINE"
U 113D, 0810,0038,0180,0978,0000,113E   ;1978 ERROR1: D_RC[0F]
U 113E, 0819,0034,4D80,0800,0000,104E   ;1979  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E   ;1980 104E: D_D.OR.K[.4],J/CALLCON
            ;1981
```

```
            ;1982  .PAGE
            ;1983  .TOC "   ERROR1 WITH PARAMETER"
            ;1984 ;++
            ;1985 ; FUNCTIONAL DESCRIPTION:
            ;1986 ;
            ;1987 ; This subroutine takes  as parameter the number of arguments
            ;1988 ; to be printed to the console in the case of an error logout.
            ;1989 ;
            ;1990 ; CALLING CONSTRAINT:
            ;1991 ;
            ;1992 ; =0
            ;1993 ;INPUTS:
            ;1994 ;
            ;1995 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;1996 ;--
            ;1997 =0
            ;1998 ERR1.ARG:
U 1038, 0000,003D,0180,0800,0000,1145  ;1999  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1039, 0000,003C,0180,0800,0000,113D  ;2000  J/ERROR1  ; Go to ERROR1
            ;2001 ;
            ;2002 ;
            ;2003
```

```
          ;2004  .PAGE
          ;2005  .TOC "    ERROR2 ROUTINE"
U 113F, 0810,0038,0180,0978,0000,1140  ;2006 ERROR2: D_RC[0F]
U 1140, 0819,0034,4D80,0800,0000,105A  ;2007  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2008 105A: D_D.OR.K[.6],J/CALLCON
          ;2009
```

```
            ;2010  .PAGE
            ;2011  .TOC "   ERROR 2 WITH PARAMETER"
            ;2012 ;++
            ;2013 ; FUNCTIONAL DESCRIPTION:
            ;2014 ;
            ;2015 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2016 ; except that in this case after setting the number of arguments
            ;2017 ; too the console, control is transferred to routine ERROR2.
            ;2018 ;
            ;2019 ; INPUTS:
            ;2020 ;
            ;2021 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2022 ;
            ;2023 ;--
            ;2024 =0
            ;2025 ERR2.ARG:
U 103A, 0000.003D.0180.0800.0000.1145  ;2026  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 103B, 0000.003C.0180.0800.0000.113F  ;2027  J/ERROR2  ; Go to ERROR2
            ;2028 ;
            ;2029 ;
            ;2030
```

```
            ;2031   .PAGE
            ;2032   .TOC "    CALLCON ROUTINE"
            ;2033 105E:
            ;2034 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2035  ID[TXDB]_D.J/CALLCON  ; CALL THE CONSOLE
            ;2036
```

```
            ;2037  .PAGE
            ;2038  .TOC "   SCRATCH PAD LOCATIONS"
            ;2039 ;*
            ;2040 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2041 ;-
U 13F0, 0000,003C,0180,0800,0000,13F1   ;2042 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2   ;2043 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3   ;2044 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4   ;2045 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5   ;2046 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6   ;2047 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7   ;2048 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8   ;2049 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9   ;2050 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA   ;2051 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB   ;2052 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC   ;2053 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD   ;2054 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE   ;2055 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF   ;2056 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155   ;2057 13FF: NOP
U 1155, 0000,003C,0180,0800,0000,12AA   ;2058 1155: NOP
U 12AA, 0000,003C,0180,0800,0000,1141   ;2059 12AA: NOP
            ;2060
```

```
            ;2061  .PAGE
            ;2062  .TOC "   SUBTEST ROUTINE"
            ;2063 ;+
            ;2064 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2065 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2066 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2067 ; TYPING IT.
            ;2068 ;-
U 1141, 0810,0038,4D80,0978,0000,1142  ;2069 SUBTST: D_RC[0F],KMX/.FF00
U 1142, 0019,4016,4D80,09F8,0000,0001  ;2070  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2071
```

```
            ;2072  .PAGE
            ;2073  .TOC "    SUBTEST 1 ROUTINE"
            ;2074 ;*****************************************************************
            ;2075 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2076 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2077 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2078 ;-
            ;2079 SUBTEST1:
U 1143, 0810.0038.0180.0978.0000.1144  ;2080  D_RC[0F]
U 1144, 0019.0032.4D80.09F8.0000.0001  ;2081  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2082
```

```
            ;2083  .PAGE
            ;2084  .TOC "   SETRCF ROUTINE"
            ;2085 ;+
            ;2086 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2087 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2088 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2089 ;-
U 1145, 0010,0038,01C0,0978,0000,1146  ;2090 SETRCF: Q_RC[0F]
U 1146, 0019,2034,4DC0,0800,0000,1147  ;2091  Q_Q.AND.K[.FF00]
U 1147, 0019,2032,1D80,09F8,0000,0001  ;2092  RC[0F]_Q.OR.SC,RETURN1
            ;2093
```

```
            ;2094  .PAGE
            ;2095  .TOC "   ENBLFLT ROUTINE"
            ;2096 ;*****************************************************************
            ;2097 ;*
            ;2098 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2099 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2100 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2101 ;
            ;2102 ; D AND SC GET CLOBBERED
            ;2103 ;-
            ;2104 ;*****************************************************************
U 1148, 0F00,003C,2180,0800,0084,7149  ;2105 ENBFLT: SC_K[.14],D_0
U 1149, 0000,003C,0580,0800,0084,B14A  ;2106  SC_SC-K[.1]
U 114A, 0801,001C,0180,0800,0000,114B  ;2107  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 114B, 0600,003C,6D80,3C00,0000,114C  ;2108  ID[FAULT]_D,D_D.RIGHT
U 114C, 0000,003E,6D80,3C00,0000,0001  ;2109  ID[FAULT]_D,RETURN1
            ;2110
```

```
            ;2111  .PAGE
            ;2112  .TOC "    NOINTR ROUTINE"
            ;2113 ;*****************************************************************
            ;2114 ;*
            ;2115 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
            ;2116 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
            ;2117 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
            ;2118 ; IF ANY INTERRUPTS OCCURRED:
            ;2119 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
            ;2120 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
            ;2121 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
            ;2122 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
            ;2123 ;  RETURN1 IS USED TO GET BACK
            ;2124 ; ELSE
            ;2125 ;  RETURN2 IS USED
            ;2126 ;
            ;2127 ; THE D AND Q REGISTERS GET CLOBBERED.
            ;2128 ;
            ;2129 ;-
            ;2130 ;*****************************************************************
            ;2131 =0
U 103C, 0000,003D,0180,0800,0000,1152  ;2132 NOINTR: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 103D, 0000,003C,35F0,2C00,0000,114D  ;2133  Q_ID[VECTOR]
U 114D, 0019,2034,8180,0800,0010,114E  ;2134  ALU_Q.AND.K[.3FF],CLK.UBCC
U 114E, 0000,013C,0180,0800,0000,103E  ;2135  Z?
U 103E, 0000,003C,0180,0800,0000,1157  ;2136 =0 J/ERRFLT    ; INTR. OCCURRED,SET FLAG,GET DATA
U 103F, 0000,003E,0180,0800,0000,0002  ;2137  RETURN2     ; NO INTR, RETURN2
            ;2138 =0
U 1040, 0000,003D,0180,0800,0000,1152  ;2139 WRTOUT: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1041, 0000,003C,35F0,2C00,0000,114F  ;2140  Q_ID[VECTOR]
U 114F, 0819,2034,8180,0800,0000,1150  ;2141  ALU_Q.AND.K[.3FF],D_ALU
U 1150, 0019,0020,A580,0800,0010,1151  ;2142  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 1151, 0000,013C,0180,0800,0000,1042  ;2143  Z?
U 1042, 0000,003C,0180,0800,0000,1157  ;2144 =0 J/ERRFLT    ; WRONG VECTOR,SET FLAG,GET DATA
U 1043, 0000,003E,0180,0800,0000,0002  ;2145  RETURN2     ; OK, RETURN2
U 1152, 0F00,003C,0180,0800,0000,1153  ;2146 STROBE: D_0     ; SET LOWEST IPL ACTIVE
U 1153, 0000,003C,3D80,3C00,0000,1154  ;2147  ID[PSL]_D    ;
U 1154, 0000,003C,0180,0800,4000,1156  ;2148  IEK/ISTR    ; STROBE INTERRUPT
U 1156, 0000,003E,0180,0800,0000,0001  ;2149  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 1157, 0018,0038,0580,09A8,0000,1158  ;2150 ERRFLT: RC[5]_K[.1]    ; Set Flag
U 1158, 0001,203C,65F0,2DC0,0000,1159  ;2151  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 1159, 0001,203C,6DF0,2DB8,0000,115A  ;2152  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 115A, 0001,203E,0180,09B0,0000,0001  ;2153  RC[6]_Q,RETURN1    ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
            ;2154
            ;2155
```

```
            ;2156  .PAGE
            ;2157  .TOC "   GENREATE CONSTANT ROUTINE (DCE)"
            ;2158 ;////////////////////////////////////////////////////////////////
            ;2159 ;*
            ;2160 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2161 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2162 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2163 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2164 ;
            ;2165 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2166 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2167 ;
            ;2168 ;  =0 D_0,CALL,J/DC5
            ;2169 ;   NOP
            ;2170 ;  =0 CALL,J/DC5
            ;2171 ;-
            ;2172
U 115B, 0018,0038,C180,0800,0000,115E  ;2173 DCE: ALU_K[.FFFF],J/S10
            ;2174
U 115C, 0118,0038,1B80,0800,0000,1160  ;2175 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 115D, 0118,0038,0B80,0800,0000,1160  ;2176 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 115E, 0118,0038,0780,0800,0000,1160  ;2177 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 115F, 0118,0038,C380,0800,0000,1160  ;2178 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
U 1160, 0100,003E,0380,0800,0000,0001  ;2179 SX: D_D.LEFT2,SI/7,RETURN1 ; EXIT
            ;2180
            ;2181
```

```
            ;2182   .PAGE
            ;2183   .TOC "    DELAY ROUTINE"
U 1161. 0818,0038,0180,0800,0000,105E   ;2184 DELAY: D_K[.8],J/CALLCON
            ;2185
```

```
;2186  .PAGE
;2187  .TOC " DIAGNOSTIC SPECIFIC ROUTINES"
```

```
            ;2188  .PAGE
            ;2189  .TOC "   CHECK UTRAP"
            ;2190 ;**
            ;2191 ;FUNCTIONAL DESCRIPTION:
            ;2192 ;
            ;2193 ; This subroutine is used to check wether a Utrap occured.
            ;2194 ; It takes the contents of the Save Utrap flags register
            ;2195 ; R[0E], and compares them to the contrnts of the Utrap
            ;2196 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2197 ; according to the results of the test  (i.e., contents of
            ;2198 ; these registers are equal or not).
            ;2199 ; CALLING CONSTRAINT:
            ;2200 ;
            ;2201 ; =00
            ;2202 ;
            ;2203 ; INPUTS:
            ;2204 ;
            ;2205 ; R[0E]- Saved Utrap Mask
            ;2206 ; R[0F]- Expected Utrap Mask
            ;2207 ;
            ;2208 ;--
            ;2209 CHECK_UTRAP:
U 1162, 0800,003C,0180,0A70,0000,1163  ;2210  D_R[0E]    ; Reg D is loaded with the trap mask
U 1163, 0001,003C,0180,09C0,0000,1164  ;2211  RC[08]_D   ; Save expected Utrap flag for error logout
U 1164, 0800,003C,0180,0A78,0000,1165  ;2212  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1165, 0001,003C,0180,09B8,0000,1166  ;2213  RC[07]_D   ; Save in RC[07]
            ;2214  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1166, 000D,0020,0180,0A70,0010,1167  ;2215         CLK.UBCC
U 1167, 0003,013C,0180,0AF8,0000,1044  ;2216  Z?,R[0F]_0  ; Branch if no traps
U 1044, 0000,003E,0180,0800,0000,0002  ;2217 =0 RETURN2    ; Utrap occured
U 1045, 0000,003E,0180,0800,0000,0001  ;2218  RETURN1    ; No Utrap  return
            ;2219
```

```
            ;2220  .PAGE
            ;2221  .TOC "   Set Trap Mask"
            ;2222 ;**
            ;2223 ; FUNCTIONAL DESCRIPTION:
            ;2224 ;
            ;2225 ; This routine is used to set a bit in the Micro Trap Mask
            ;2226 ; R[0F].
            ;2227 ;
            ;2228 ; CALLING CONSTRAINT:
            ;2229 ;
            ;2230 ; =0
            ;2231 ;
            ;2232 ; INPUTS:
            ;2233 ;
            ;2234 ; SC - Contains bit number to set.
            ;2235 ;
            ;2236 ; OUTPUTS:
            ;2237 ;
            ;2238 ; rc[0E] - Contains the current trap mask
            ;2239 ;
            ;2240 ; SIDE EFFECTS:
            ;2241 ;
            ;2242 ; d regsiter is destroyed
            ;2243 ;--
            ;2244 SET_TRAP_MASK:
U 1168, 0800,003C,0180,0A78,0000,1169  ;2245  d_r[0f]
U 1169, 0801,001C,0180,0AF8,0000,116A  ;2246  r[0f]_d.ornot.mask,d_alu ; Set mask
U 116A, 0001,003E,0180,0AF0,0000,0001  ;2247  r[0E]_d,return1    ; Save mask and return
            ;2248
```

```
            ;2249  .PAGE
            ;2250  .TOC "   Disable Cache Routine"
            ;2251 ;**
            ;2252 ; FUNCTIONAL DESCRIPTION:
            ;2253 ;
            ;2254 ; This routine disables cache hits by setting bits <16:15>
            ;2255 ; in the maintenance register.
            ;2256 ;
            ;2257 ; CALLING SEQUENCE:
            ;2258 ;
            ;2259 ; =0
            ;2260 ;
            ;2261 ; SIDE EFFECTS:
            ;2262 ;
            ;2263 ; D & Q Registers destroyed
            ;2264 ;--
            ;2265 DISABLE.CACHE:
U 116B, 0818,0038,4580,0800,0000,116C  ;2266  d_k[.8000]  ; ... and seed constant
U 116C, 0500,003C,0180,0800,0000,116D  ;2267  d_d.left  ; Generate final constant
U 116D, 0819,0030,4580,0800,0000,116E  ;2268  d_d.or.k[.8000]  ; ... = 00018000
U 116E, 0000,003E,7580,3C00,0000,0001  ;2269  id[maint]_d,return1 ; Disable cache and return
            ;2270
```

```
            ;2271  .PAGE
            ;2272  .TOC "   Initialize the SBI"
            ;2273 ;++
            ;2274 ; FUNCTIONAL DESCRIPTION:
            ;2275 ;
            ;2276 ; This routine performs the following functions:
            ;2277 ;
            ;2278 ;  Executes an SBI Unjam
            ;2279 ;  Clears the SBI Fault Latch
            ;2280 ;  Clears the SBI Error Register
            ;2281 ;
            ;2282 ; CALLING CONSTRAINT:
            ;2283 ;
            ;2284 ; =0
            ;2285 ;
            ;2286 ; SIDE EFFECTS:
            ;2287 ;
            ;2288 ; D & Q Register destroyed
            ;2289 ;--
            ;2290 =0
            ;2291 SBI.INIT:
U 1046, 0000,003D,0180,0800,0000,11BD  ;2292  call[sbi.unjam]  ; Unjam the SBI
U 1047, 0818,0038,C180,0800,0000,116F  ;2293  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 116F, 0801,0028,6580,3C00,0000,1170  ;2294  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1170, 0F00,003C,6D80,3C00,0000,1171  ;2295  id[fault]_d,d_0
U 1171, 0000,003C,6D80,3C00,0000,1172  ;2296  id[fault]_d
U 1172, 0000,003E,6580,3C00,0000,0001  ;2297  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2298
            ;2299
```

```
            ;2300  .PAGE
            ;2301  .TOC "   FIND MS780-E MEMORY"
            ;2302 ;++
            ;2303 ; FUNCTIONAL DESCRIPTION:
            ;2304 ;
            ;2305 ;          The diagnostic is designed to handle up to 4 (four)
            ;2306 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2307 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2308 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2309 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2310 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2311 ;     first MS780-E found will have its starting address set to 0
            ;2312 ;     (zero).
            ;2313 ;     All the other MS780-E/H's found will have their starting address
            ;2314 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2315 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2316 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2317 ;     current MS780-E unit under test.
            ;2318 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
            ;2319 ;     set to maximum.
            ;2320 ;
            ;2321 ; CALLING CONSTRAINT:
            ;2322 ;
            ;2323 ; =00
            ;2324 ;
            ;2325 ; OUTPUTS:
            ;2326 ;
            ;2327 ; rc[0e] - Contains address of Configuration Register
            ;2328 ; r[0]  - Contains TR level
            ;2329 ;
            ;2330 ; SIDE EFFECTS:
            ;2331 ;
            ;2332 ; D register is destroyed.
            ;2333 ;--
            ;2334 =0
            ;2335 FIND.MS780E:
U 1048, 0000,003D,0180,0800,0000,1046  ;2336  call[sbi.init]  ; Make sure SBI is enabled
U 1049, 0003,003C,0180,0988,0000,1173  ;2337  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1173, 0818,0038,1980,0800,0000,1174  ;2338  d_k[zero]  ; Clear MS780-E/H counter
U 1174, 0000,003C,F980,3C00,0000,1175  ;2339  id[3e]_d  ; ..
U 1175, 0018,0038,0580,0A80,0000,1176  ;2340  r[0]_k[.1]  ; Init TR counter
U 1176, 0F03,003C,0180,09F0,0000,104A  ;2341  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2342      ; ...CNFG A Reg
            ;2343 =0
            ;2344 FIND.MEM.LOOP:
U 104A, 0800,003D,0180,0A00,0000,119C  ;2345  d_r[0],call[generate.io]; Generate address of TR level
U 104B, 0001,003C,0180,09F0,0200,104C  ;2346  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 104C, 0000,003D,8980,0800,0084,7168  ;2347 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2348  d[long]_cache.p, ; Read the specified tr level
U 104D, 0000,003C,0180,C970,0000,1177  ;2349   ic_rc[0e]  ; ...
U 1177, 0000,003C,0180,0A78,0010,1178  ;2350  alu_r[0f],clk.ubcc ; Check for no response
U 1178, 0001,013C,0180,0880,0082,1050  ;2351  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 1050, 0000,003C,0180,0800,0000,1179  ;2352 =0 j/check.adpt.code ; Check for a memory
U 1051, 0000,003C,0180,0800,0000,1198  ;2353  j/find.mem.lp1  ; Try next tr
            ;2354 CHECK.ADPT.CODE:
```

```
U 1179, 0000,003C,1D80,0800,1404,717A  ;2355  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 117A, 0000,163C,0180,0800,0000,1008  ;2356  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,717B  ;2357 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,717C  ;2358  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,1198  ;2359  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,1198  ;2360  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7181  ;2361  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,1198  ;2362  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1185  ;2363  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1185  ;2364  rc[0e]_lc,j/found.ms780e ; MS780-E
           ;2365   ;
           ;2366   ; Found an MS780-A or -C. Set the starting address
           ;2367   ; to maximum.
           ;2368   ;
U 117B, 0818,0038,5D80,0800,0000,117D  ;2369 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 117C, 0818,0038,0580,0800,0000,117D  ;2370 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
           ;2371 MS780.COMMON:
U 117D, 0819,0030,7180,0800,0000,117E  ;2372  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 117E, 0000,003C,01F8,0803,0080,D17F  ;2373  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 117F, 0D00,003C,0180,0800,0000,1180  ;2374  d_dal.sc  ; Put data in bits<29:14>
           ;2375  cache.p_d[long], ; Set the starting address to maximum
U 1180, 0000,003C,0180,A800,0000,1198  ;2376       j/find.mem.lp1 ; and continue TR scan
           ;2377   ;
           ;2378   ; Found an MA780. Set the starting address to maximum
           ;2379   ;
           ;2380 MA780.MAX:
U 1181, 0818,0038,39F8,0803,0000,1182  ;2381  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1182, 0D00,003C,0180,0803,0000,1183  ;2382  d_dal.sc,va_va+4 ; Put in proper position
U 1183, 0000,003C,0180,0803,0000,1184  ;2383  va_va+4   ; Point to Port Invalidate Ctrl Register
           ;2384  cache.p_d[long], ; Set starting address to maximum
U 1184, 0000,003C,0180,A800,0000,1198  ;2385       j/find.mem.lp1
           ;2386 ;
           ;2387 ; Found an MS780E
           ;2388 ;
           ;2389 FOUND.MS780E:
U 1185, 0000,003C,F9F0,2C00,0000,1186  ;2390  q_id[3e]  ; Set up to see how many MS780-E's
           ;2391  alu_q.xor.k[.3], ; Are there Maximum units?
U 1186, 0019,2020,0D80,0800,0010,1187  ;2392   clk.ubcc ; Check condition codes
U 1187, 0000,013C,0180,0800,0000,1052  ;2393  z?    ; Are we at maximum units for system
U 1052, 0000,003C,0180,0800,0000,1188  ;2394 =0 J/ms780e.tst  ; No go find how many units ther are
U 1053, 0818,0038,C180,0800,0000,1054  ;2395  d_k[.ffff]  ; Yes set address to maximum
U 1054, 0000,003D,0180,0800,0000,11A0  ;2396 =0 call[set.start.addr] ; .....
U 1055, 0000,003C,0180,0800,0000,1198  ;2397  j/find.mem.lp1  ; Go check to see if we are done
           ;2398 ;
           ;2399 MS780E.TST:
           ;2400  alu_rc[01],  ; Check "Found MS780E Flag"
U 1188, 0010,0038,0180,0908,0010,1189  ;2401   clk.ubcc ; Check condition codes
U 1189, 0810,0138,0180,0970,0000,1056  ;2402  z?,d_rc[0e]  ; Is this unit the first MS780E ?
           ;2403 =0 j/not.first.ms780e, ; No
U 1056, 0818,0038,C180,0800,0000,105C  ;2404   d_k[.ffff] ; Set starting address
U 1057, 0001,003C,0180,0988,0000,118A  ;2405  rc[01]_d  ; Yes put base address in rc[01]
U 118A, 0818,0038,7980,0800,0000,118B  ;2406  d_k[.30]  ; Set up SC to point to first unit
U 118B, 0019,0014,F580,0800,0082,118C  ;2407  alu_d+k[.a],sc_alu ; ...
U 118C, 0810,0038,0180,0908,0000,118D  ;2408  d_rc[01]  ; Put base address of unit in D
U 118D, 0000,003C,0180,3400,0000,118E  ;2409  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 118E, 0F00,003C,0180,0800,0000,1058  ;2410  d_0   ; Prepare to set starting address to Zero
U 1058, 0000,003D,0180,0800,0000,11A0  ;2411 =0 call[set.start.addr] ; Go do it
        ;2412  j/find.mem.lp1,  ; Check to see if we are done
U 1059, 0003,003C,0180,09E8,0000,1198  ;2413   rc[0d]_0 ; ....
        ;2414 =0
        ;2415 NOT.FIRST.MS780E:
U 105C, 0000,003D,0180,0800,0000,11A0  ;2416  call[set.start.addr] ; Go set starting address to maximum
U 105D, 0000,003C,F9F0,2C00,0000,118F  ;2417  q_id[3e]  ; Get the number of MS780-Es found
U 118F, 0819,2014,0580,0800,0000,1190  ;2418  d_q+k[.1]  ; Increment this counter
U 1190, 0000,003C,F980,3C00,0000,1191  ;2419  id[3e]_d  ; Save the counter
U 1191, 0018,0038,11C0,0800,0000,1192  ;2420  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2421      ; ...were the I/O base addresses will be stored
U 1192, 081D,2000,7980,0800,0000,1193  ;2422  d_q-d,k[.30]  ; ... adjust pointer
U 1193, 0819,0014,7980,0800,0000,1194  ;2423  d_d+k[.30]  ; Point the SC to the ID register
U 1194, 0000,003C,F580,0800,0000,1195  ;2424  k[.a]   ; ...to receive I/O base address
U 1195, 0019,0014,F580,0800,0082,1196  ;2425  alu_d+k[.a],sc_alu ; ...
U 1196, 0810,0038,0180,0970,0000,1197  ;2426  d_rc[0e]  ; Get base address to d
U 1197, 0000,003C,0180,3400,0000,1198  ;2427  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2428   ;
        ;2429   ; Try next tr
        ;2430   ;
        ;2431 FIND.MEM.LP1:
U 1198, 0818,0014,0580,0A80,0000,1199  ;2432  r[0]_ia+k[.1],d_alu ; Increment TR level
        ;2433  alu_d.and.k[.f],
U 1199, 0019,0034,6180,0800,0010,119A  ;2434      clk.ubcc  ; Check if all done
U 119A, 0000,013C,0180,0800,0000,1060  ;2435  z?   ; Branch if yes
U 1060, 0000,003C,0180,0800,0000,104A  ;2436 =0 j/find.mem.loop  ; Try next TR
U 1061, 0810,0038,0180,0908,0010,119B  ;2437  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 119B, 0000,013C,0180,0800,0000,1062  ;2438  z?   ; Check condition codes
U 1062, 0001,003E,0180,09F0,0000,0002  ;2439 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1063, 0000,003E,0180,0800,0000,0001  ;2440  return1   ; No MS780E found
        ;2441
```

```
            ;2442  .PAGE
            ;2443  .TOC "   Generate IO Address"
            ;2444 ;**
            ;2445 ; FUNCTIONAL DESCRIPTION:
            ;2446 ;
            ;2447 ; This routine is used to generate an SBI Configuration Register
            ;2448 ; address from a TR level.
            ;2449 ;
            ;2450 ; CALLING CONSTRAINT:
            ;2451 ;
            ;2452 ; =0
            ;2453 ;
            ;2454 ; INPUTS:
            ;2455 ;
            ;2456 ; D - Contains TR level
            ;2457 ;
            ;2458 ; OUTPUTS:
            ;2459 ;
            ;2460 ; D - Contains SBI IO address
            ;2461 ;--
            ;2462 GENERATE.IO:
U 119C, 0000,003C,89F8,0800,0084,719D  ;2463  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 119D, 0D00,003C,8D80,0800,0084,719E  ;2464  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 119E, 0000,003C,0980,0800,0084,B19F  ;2465  sc_sc-k[.2]  ; insert bit 29
U 119F, 0801,001E,0180,0800,0000,0001  ;2466  d_d.ornot.mask,return1 ; and return
            ;2467
            ;2468
```

```
            ;2469  .PAGE
            ;2470  .TOC "   Set Starting Address"
            ;2471 ;++
            ;2472 ; FUNCTIONAL DESCRIPTION:
            ;2473 ;
            ;2474 ; This routine is used to set the starting address of the
            ;2475 ; memory to the specified value.
            ;2476 ;
            ;2477 ; CALLING CONSTRAINT:
            ;2478 ;
            ;2479 ; =0
            ;2480 ;
            ;2481 ; INPUTS:
            ;2482 ;
            ;2483 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2484 ;     (B Register Image divided by 2**16)
            ;2485 ; rc[0] - Contains Address of Register A
            ;2486 ;
            ;2487 ; OUTPUTS:
            ;2488 ;
            ;2489 ; d - Contains aligned starting address (B Register Image)
            ;2490 ;
            ;2491 ; SIDE EFFECTS:
            ;2492 ;
            ;2493 ; d,q,va, and sc are destroyed
            ;2494 ;--
            ;2495 SET.START.ADDR:
U 11A0, 0010,0038,6580,0970,0284,71A1  ;2496  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11A1, 0000,003C,01F8,0803,0000,11A2  ;2497  va_va+4,q_0  ; Set address to Register B
            ;2498  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11A2, 0D00,003C,0980,0800,0084,B1A3  ;2499       sc_sc-k[.2]  ; Setup to insert bit 14
U 11A3, 0801,001C,0180,0800,0000,11A4  ;2500  d_d.ornot.mask  ; Insert Write Enable bit
U 11A4, 0000,003E,0180,A800,0000,0001  ;2501  cache.p_d[long],return1 ; Set the starting address and return
            ;2502
```

```
          ;2503  .PAGE
          ;2504  .TOC "   ENABLE NEXT MS780-E"
          ;2505 ;**
          ;2506 ; FUNCTIONAL DESCRIPTION:
          ;2507 ;
          ;2508 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
          ;2509 ;    They will be tested  separately, according to the TR level at which
          ;2510 ;    they are located,  starting with the one  at the lowest level first.
          ;2511 ;          When  a test has finished  with the first unit, it will have to call
          ;2512 ;    this specific  routine  to see  if any other MS780-E unit is present
          ;2513 ;    on the SBI. If more units are  present, the first one  will be  dis-
          ;2514 ;    abled  by  setting  its  starting  address to maximum, the next unit
          ;2515 ;    will be  enabled  by  setting its starting address to 0 and the test
          ;2516 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
          ;2517 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
          ;2518 ;    ID registers 3A through 3D and  a count of the Total number of units
          ;2519 ;    found - 1 in register 3E. In this subroutine the SC register is used
          ;2520 ;    as  a  pointer  to  ID  registers 3A through 3D.
          ;2521 ;
          ;2522 ; CALLING CONSTRAINT:
          ;2523 ;
          ;2524 ;  =00
          ;2525 ;
          ;2526 ;--
          ;2527 ENABLE.NEXT.MS780E:
U 11A5, 0000,003C,F9F0,2C00,0000,11A6  ;2528  Q_ID[3E]  ; Get total number of MS780E to Q
          ;2529  ALU_Q,   ; Check to see if it is zero
U 11A6, 0001,203C,0180,0800,0010,11A7  ;2530     CLK.UBCC  ; Check Condition Codes
U 11A7, 0000,013C,0180,0800,0000,1064  ;2531  Z?   ; ...for zero
U 1064, 0000,003C,0180,0800,0000,11A8  ;2532 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1065, 0000,003E,0180,0800,0000,0002  ;2533  RETURN2   ; Zero No more units to test
          ;2534 ENABLE.NEXT:
U 11A8, 0818,0038,C180,0800,0000,1066  ;2535  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1066, 0000,003D,0180,0800,0000,11A0  ;2536 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1067, 0818,0038,7980,0800,0000,11A9  ;2537  D_K[.30]  ; Prepare to point to the ID register holding
          ;2538     ; ...the I/O Base address of the next MS780-E
U 11A9, 0819,0014,1180,0800,0000,11AA  ;2539  D_D+K[.4]  ; ...
U 11AA, 0000,003C,F580,0800,0000,11AB  ;2540  K[.A]   ; ...
U 11AB, 0819,0014,F580,0800,0000,11AC  ;2541  D_D+K[.A]  ; ...
U 11AC, 0000,003C,F9F0,2C00,0000,11AD  ;2542  Q_ID[3E]  ; Get MS780-E Counter
          ;2543  D_D-Q,   ; D now holds the pointer to the ID register
U 11AD, 081D,0000,0180,0800,0082,11AE  ;2544   SC_ALU  ; ...
U 11AE, 0000,003C,01F0,2400,0000,11AF  ;2545  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11AF, 0001,203C,0180,09F0,0000,11B0  ;2546  RC[0E]_Q  ; Save it also in RC[0E]
U 11B0, 0F03,003C,0180,09E8,0000,1068  ;2547  RC[0D]_0,D_0  ; Set starting address to zero
U 1068, 0000,003D,0180,0800,0000,11A0  ;2548 =0 CALL[SET.START.ADDR] ; ....
U 1069, 0F00,003C,F9F0,2C00,0000,11B1  ;2549  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11B1, 081D,2008,0180,0800,0000,11B2  ;2550  D_Q-D-1   ; Subtract 1 from total
U 11B2, 0000,003C,F980,3C00,0000,106A  ;2551  ID[3E]_D  ; Adjust the pointer
U 106A, 0000,003D,0180,0800,0000,11EB  ;2552 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 106B, 0000,003E,0180,0800,0000,0001  ;2553  RETURN1   ; Tell caller another unit was found
```

```
            ;2554  .PAGE
            ;2555  .TOC "   Select Controller"
            ;2556 ;++
            ;2557 ; FUNCTIONAL DESCRIPTION:
            ;2558 ;
            ;2559 ; This routine is used to put the memory system into the
            ;2560 ; specified configuration with respect to controller select.
            ;2561 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2562 ; a RETURN2 is made.
            ;2563 ;
            ;2564 ; CALLING CONSTRAINT:
            ;2565 ;
            ;2566 ; =00
            ;2567 ;
            ;2568 ; INPUTS:
            ;2569 ;
            ;2570 ; d - Contains value to write to bits<2:0> of Register A
            ;2571 ; rc[0e] - Address of Register A
            ;2572 ;
            ;2573 ; SIDE EFFECTS:
            ;2574 ;
            ;2575 ; sc,d,va are destroyed
            ;2576 ;--
            ;2577 SELECT.CNTRLR:
U 11B3, 0010.0038.0180.0970.0284.71B4  ;2578  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11B4, 0801.001C.0180.0800.0000.11B5  ;2579  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11B5, 0000.003C.0180.A800.0000.11B6  ;2580  cache.p_d[long]  ; Write the configuration Register
U 11B6, 0818.0038.5D80.0800.0000.11B7  ;2581  d_k[.7]   ; Get Misconfiguration bits
U 11B7, 0000.003C.61F8.0800.0084.71B8  ;2582  sc_k[.F],q_0  ; ...
U 11B8, 0D00.003C.0180.0800.0000.11B9  ;2583  d_dal.sc  ; ... D = x38000
U 11B9, 0000.003C.01E0.0800.0000.11BA  ;2584  q_d   ; Save mask
U 11BA, 0000.003C.0180.C800.0000.11BB  ;2585  d[long]_cache.p  ; Read CNFG A Reg and
            ;2586  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11BB, 001D.2034.0180.0800.0010.11BC  ;2587   clk.ubcc ; ...
U 11BC, 0000.013C.0180.0800.0000.106C  ;2588  z?   ; Branch if no
U 106C, 0000.003E.0180.0800.0000.0001  ;2589 =0 RETURN1   ; Bad configuration
U 106D, 0000.003E.0180.0800.0000.0002  ;2590  RETURN2   ; Configuration ok
            ;2591
```

```
;2592  .PAGE
;2593  .TOC "   SELECT LOWER CONTROLLER"
;2594 ;*************************************************************
;2595 ;++
;2596 ;
;2597 ; Functional Description:
;2598 ; ----------------------
;2599 ;
;2600 ; This subroutine can be called to select the lower
;2601 ; Controller on a MS780-E/H.
;2602 ; If the Lower controller is misconfigured then a
;2603 ; RETURN1 will be made.  Otherwise a RETURN2
;2604 ;
;2605 ; Calling Constraint: =00
;2606 ; ------------------
;2607 ;
;2608 ;
;2609 ; Inputs:
;2610 ; ------
;2611 ;
;2612 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2613 ;
;2614 ; Outputs:
;2615 ; -------
;2616 ;
;2617 ; RC[0D] will have the address of CNFG Register C
;2618 ;        ( 2000x008 )
;2619 ;
;2620 ; Side Effects:
;2621 ; ------------
;2622 ;
;2623 ; Contents of the following registers will lost:
;2624 ;
;2625 ; D
;2626 ;
;2627 ; The Lower controller on the MS780-E/H will be configured
;2628 ; when the subroutine is exitted with a RETURN2
;2629 ;--
;2630 ;*************************************************************
;2631 =00
;2632 SELECT.LOWER:
;2633  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,11B3  ;2634   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2635  RETURN1    ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2636  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2637  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2638   RETURN2   ; Let caller know that the controller
;2639       ; ...was configured properly
```

```
            ;2640  .PAGE
            ;2641  .TOC "    SELECT UPPER CONTROLLER"
            ;2642 ;*************************************************************
            ;2643 ;++
            ;2644 ;
            ;2645 ; Functional Description:
            ;2646 ; ----------------------
            ;2647 ;
            ;2648 ; This subroutine can be called to select the upper
            ;2649 ; Controller on a MS780-E/H.
            ;2650 ; If the Upper controller is misconfigured then a
            ;2651 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2652 ;
            ;2653 ; Calling Constraint: =00
            ;2654 ; ------------------
            ;2655 ;
            ;2656 ;
            ;2657 ; Inputs:
            ;2658 ; ------
            ;2659 ;
            ;2660 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2661 ;
            ;2662 ; Outputs:
            ;2663 ; -------
            ;2664 ;
            ;2665 ; RC[0D] will have the address of CNFG Register D
            ;2666 ;        ( 2000x010 )
            ;2667 ;
            ;2668 ; Side Effects:
            ;2669 ; ------------
            ;2670 ;
            ;2671 ; Contents of the following registers will lost:
            ;2672 ;
            ;2673 ; D
            ;2674 ;
            ;2675 ; The Upper controller on the MS780-E/H will be configured
            ;2676 ; when the subroutine is exitted with a RETURN2
            ;2677 ;--
            ;2678 ;*************************************************************
            ;2679 =00
            ;2680 SELECT.UPPER:
            ;2681  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,11B3  ;2682   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2683  RETURN1     ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2684  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2685  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2686   RETURN2    ; Let caller know that the controller
            ;2687       ; ...was configured properly
            ;2688
```

```
            ;2689  .PAGE
            ;2690  .TOC "   CLRSBI ROUTINE"
            ;2691 CLRSBI:
            ;2692 ;=================================================================
            ;2693 ;/////////////////////////////////////////////////////////////////
            ;2694 ;*
            ;2695 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2696 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2697 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2698 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2699 ;-
            ;2700 SBI.UNJAM:
U 11BD. 0818,0038,6580,8000,0010,106E  ;2701 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2702 =0
U 106E. 0819,0100,0580,8000,0010,106E  ;2703 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 106F. 0818,0038,6580,8800,0010,1070  ;2704  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2705 =0
U 1070. 0819,0100,0580,8800,0010,1070  ;2706 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 1071. 0818,0038,6580,8000,0010,1072  ;2707  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2708 =0
U 1072. 0819,0100,0580,8000,0010,1072  ;2709 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 1073. 0000,003E,0180,0800,0000,0001  ;2710  RETURN1    ; DONE.
            ;2711
```

```
            ;2712  .PAGE
            ;2713  .TOC "    SLOLCK ROUTINE"
            ;2714 ;*****************************************************************
            ;2715 ;*
            ;2716 ;       THIS ROUTINE PLACES THE LAST 15 LOCATIONS OF THE SILO
            ;2717 ; IN RC[0E] THRU RC[0].
            ;2718 ;
            ;2719 ;  THE STATE IN WHICH THE CALL IS DONE, AND
            ;2720 ;  THE PRECEDING STATE MUST LOOK EXACTLY AS FOLLOWS:
            ;2721 ;
            ;2722 ;   ID[COMP]_D,...
            ;2723 ;   CALL,J/SLOLCK,MCT/XXXXX
            ;2724 ;
            ;2725 ;   WHERE XXXXX IS ANY REFERENCE.
            ;2726 ;
            ;2727 ;-
            ;2728 ;*****************************************************************
            ;2729 =0
U 1074, 0818,0038,1180,0800,0000,1075   ;2730 SLOLCK: D_K[.4]    ; PROCEED TO WAIT A TOTAL OF 16 CYCLES
U 1075, 0001,003C,0180,0800,0010,11BE   ;2731 L1: ALU_D,CLK.UBCC  ; ...
U 11BE, 0819,0100,0580,0800,0000,1076   ;2732  Z?,D_D-K[.1]
U 1076, 0000,003C,0180,0800,0000,1075   ;2733 =0 J/L1    ; ...
U 1077, 0000,003C,6180,0800,0084,71BF   ;2734  SC_K[.F]  ; START FOR PLACING SILO IN RC REGISTERS
U 11BF, 0000,003C,0580,0800,0084,B1C0   ;2735 L2: SC_SC-K[.1]  ; STARTING WITH RC[0E]
U 11C0, 0000,003C,61F0,2C00,0000,11C1   ;2736  Q_ID[SILO]
U 11C1, 0001,203C,0180,0838,0000,11C2   ;2737  RC(SC)_ALU,ALU_Q
U 11C2, 0018,0038,1D80,0800,0010,11C3   ;2738  ALU_SC,CLK.UBCC
U 11C3, 0000,013C,0180,0800,0000,1078   ;2739  Z?
U 1078, 0000,003C,0180,0800,0000,11BF   ;2740 =0 J/L2
U 1079, 0F00,003C,0180,0800,0000,11C4   ;2741  D_0
U 11C4, 0000,003E,7180,3C00,0000,0001   ;2742  ID[COMP]_D,RETURN1 ; UNLOCK SILO
            ;2743
```

```
            ;2744  .PAGE
            ;2745  .TOC "   CLCPTO ROUTINE"
            ;2746 ;*************************************************************
            ;2747 ;* THIS ROUTINE CLEARS THE SBI.ERR REGISTER
            ;2748 ;-
U 11C5, 081B,0000,0580,0800,0000,11C6  ;2749 CLCPTO: D_0-K[.1]
U 11C6, 0801,0028,6580,3C00,0000,11C7  ;2750  ID[SBI.ERR]_D,D_NOT.D
U 11C7, 0000,003E,6580,3C00,0000,0001  ;2751  ID[SBI.ERR]_D,RETURN1
            ;2752
```

```
            ;2753   .PAGE
            ;2754   .TOC "    INIT ROUTINE"
            ;2755   ;=============================================================================
            ;2756   ;
            ;2757   ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2758   ;    PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2759   ;     FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2760   ;
            ;2761   ;=============================================================================
            ;2762   ;
U 11C8. 0003.003C.0180.0AF8.0000.107A   ;2763 INIT: R[0F]_0     ;CLEAR U-TRAPS
U 107A. 0000.003D.0180.0800.0000.11BD   ;2764 =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 107B. 0000.003C.0180.0800.0000.107C   ;2765  NOP
U 107C. 0000.003D.0180.0800.0000.11D7   ;2766 =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2767           ;  INTRUPT REG
U 107D. 0000.003C.0180.0800.0000.107E   ;2768  NOP
U 107E. 0000.003D.0180.0800.0000.11DE   ;2769 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 107F. 0000.003C.0180.0800.0000.1080   ;2770  NOP
U 1080. 0000.003D.0180.0800.0000.11E2   ;2771 =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 1081. 0000.003C.0180.0800.0000.1082   ;2772  NOP
U 1082. 0000.003D.0180.0800.0000.11DA   ;2773 =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 1083. 0000.003C.0180.0800.0000.1084   ;2774  NOP
U 1084. 0000.003D.0180.0800.0000.11E6   ;2775 =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 1085. 0000.003C.0180.0800.0000.1086   ;2776  NOP
U 1086. 0000.003D.0180.0800.0000.1148   ;2777 =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 1087. 0818.0038.4580.0800.0000.1088   ;2778  D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;2779 =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 1088. 0000.003D.7980.3C00.0000.116B   ;2780   ID[PARITY]_D
U 1089. 0F00.003C.0180.0800.0000.11C9   ;2781  D_0
U 11C9. 0000.003C.4980.3C00.0000.11CA   ;2782  ID[TBER0]_D    ;SHUT TB OFF
U 11CA. 0000.003E.7180.3C00.0000.0001   ;2783  ID[COMP]_D.RETURN1  ;DISABLE SILO INTERRUPTS
            ;2784
```

```
            ;2785   .PAGE
            ;2786   .TOC "    CHECK FOR INTERRUPT VECTORS"
            ;2787   ;
            ;2788   ;=============================================================================
            ;2789   ;*
            ;2790   ;CHECK FOR INTERRUPTS PENDING
            ;2791   ;
            ;2792   ; INTERRUPT  VECTOR CHECKED
            ;2793   ; ------------------------------------
            ;2794   ; WRITE TIMEOUT  ^X60
            ;2795   ; SBI FAULT  ^X5C
            ;2796   ; CRD,RDS TAG  ^X54
            ;2797   ; SILO    ^X50
            ;2798   ;
            ;2799   ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2800   ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2801   ;
            ;2802   ; DATA: EXPECTED VECTOR
            ;2803   ;  VECTOR STROBED
            ;2804   ;-
            ;2805   ;=============================================================================
            ;2806   ;
            ;2807 SBIFLT: ;=============================================================================
            ;2808   ;CHECKS FOR THE SBI FAULT INTERRUPT, ^X5C
            ;2809   ;=============================================================================
            ;2810   ;
U 11CB, 0818,0038,3580,0800,4000,11CC   ;2811   IEK/ISTR,D_K[.50]  ;STROBE THE VECTOR
U 11CC, 0819,0030,8580,0800,0000,11CD   ;2812   D_D.OR.K[.C]   ;GENERATE THE EXPECTED VECTOR
U 11CD, 0000,003C,35F0,2C00,0000,11CE   ;2813   Q_ID[VECTOR]
U 11CE, 0019,2034,81C0,0800,0000,11CF   ;2814   Q_Q.AND.K[.3FF]
U 11CF, 001D,0020,0180,0800,0010,11D0   ;2815   ALU_D[XOR]Q,CLK.UBCC
U 11D0, 0000,013C,0180,0800,0000,108A   ;2816   Z?
U 108A, 0000,003C,0180,0800,0000,1157   ;2817  =0 J/ERRFLT    ;NOT THE EXPECTED INTERRUPT
U 108B, 0000,003E,0180,0800,0000,0002   ;2818   RETURN2
            ;2819 SILINT: ;=============================================================================
            ;2820   ;CHECK FOR SILO INTERRUPT VECTOR    ^X50
            ;2821   ;=============================================================================
            ;2822   ;
U 11D1, 0000,003C,0180,0800,4000,11D2   ;2823   IEK/ISTR    ;STROBE THE VECTOR
U 11D2, 0818,0038,3580,0800,0000,11D3   ;2824   D_K[.50]
U 11D3, 0000,003C,35F0,2C00,0000,11D4   ;2825   Q_ID[VECTOR]
U 11D4, 0019,2034,81C0,0800,0000,11D5   ;2826   Q_Q.AND.K[.3FF]    ;MASK
U 11D5, 001D,0020,0180,0800,0010,11D6   ;2827   ALU_D[XOR]Q,CLK.UBCC   ;COMPARE
U 11D6, 0000,013C,0180,0800,0000,108C   ;2828   Z?
U 108C, 0000,003C,0180,0800,0000,1157   ;2829  =0 J/ERRFLT    ;WRONG VECTOR
U 108D, 0000,003E,0180,0800,0000,0002   ;2830   RETURN2     ;O.K.,NORMAL RETURN
            ;2831
```

```
            ;2832  .PAGE
            ;2833  .TOC "   SETPSL ROUTINE"
            ;2834 SETPSL: ;=================================================================
            ;2835  ;
            ;2836  ;DROP THE CPU IPR LEVEL TO
            ;2837  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2838  ; PENDING
            ;2839  ;
            ;2840  ;=================================================================
            ;2841  ;
U 11D7, 0F00,003C,0180,0800,0000,11D8  ;2842  D_0
U 11D8, 0000,003C,3980,3C00,0000,11D9  ;2843  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 11D9, 0000,003E,3D80,3C00,0000,0001  ;2844  RETURN1,ID[PSL]_D
            ;2845
```

```
            ;2846   .PAGE
            ;2847   .TOC "    CLRTIM ROUTINE"
            ;2848 CLRTIM: ;=================================================================
            ;2849   ;
            ;2850   ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2851   ; SBI ERROR REGISTER
            ;2852   ;
            ;2853   ;=================================================================
            ;2854   ;
U 11DA, 0818,0038,0580,0800,0000,11DB   ;2855  D_K[.1]
U 11DB, 0000,003C,85F8,0800,0084,71DC   ;2856  Q_0,SC_K[.C]
U 11DC, 0D00,003C,0180,0800,0000,11DD   ;2857  D_DAL,SC
U 11DD, 0000,003E,6580,3C00,0000,0001   ;2858  ID[SBI.ERR]_D,RETURN1
            ;2859
```

```
            ;2860   .PAGE
            ;2861   .TOC "    CLRFLT ROUTINE"
            ;2862 CLRFLT: ;=================================================================
            ;2863   ;
            ;2864   ;CLEAR THE CP FAULT BIT IN THE
            ;2865   ; FAULT REGISTER
            ;2866   ;
            ;2867   ;=================================================================
            ;2868   ;
U 11DE. 0818,0038,0180,0800,0000,11DF   ;2869  D_K[.8]
U 11DF. 0000,003C,65F8,0800,0084,71E0   ;2870  Q_0,SC_K[.10]
U 11E0. 0D00,003C,0180,0800,0000,11E1   ;2871  D_DAL.SC
U 11E1. 0000,003E,6D80,3C00,0000,0001   ;2872  ID[FAULT]_D,RETURN1
            ;2873
```

```
            ;2874   .PAGE
            ;2875   .TOC "    CLRECC ROUTINE"
            ;2876 CLRECC: ;=================================================================
            ;2877   ;
            ;2878   ;CLEAR CRD,RDS BIT IN THE
            ;2879   ;SBI ERROR REGISTER
            ;2880   ;
            ;2881   ;=================================================================
            ;2882   ;
U 11E2. 0818,0038,0D80,0800,0000,11E3   ;2883  D_K[.3]
U 11E3. 0000,003C,89F8,0800,0084,71E4   ;2884  Q_0,SC_K[.D]
U 11E4. 0D00,003C,0180,0800,0000,11E5   ;2885  D_DAL,SC
U 11E5. 0000,003E,6580,3C00,0000,0001   ;2886  ID[SBI.ERR]_D,RETURN1
            ;2887
```

```
            ;2888   .PAGE
            ;2889   .TOC "    ENBECC ROUTINE"
            ;2890 ENBECC: ;=================================================================
            ;2891   ;
            ;2892   ;ENABLE CRD,RDS INTERRUPTS
            ;2893   ;
            ;2894   ;=================================================================
            ;2895   ;
U 11E6. 0818.0038.0580.0800.0000.11E7   ;2896  D_K[.1]
U 11E7. 0000.003C.61F8.0800.0084.71E8   ;2897  Q_0,SC_K[.F]
U 11E8. 0D00.003C.0180.0800.0000.11E9   ;2898  D_DAL.SC
U 11E9. 0000.003E.6580.3C00.0000.0001   ;2899  ID[SBI.ERR]_D,RETURN1
            ;2900
```

```
            ;2901  .PAGE
            ;2902  .TOC "    Wait Loop Routine"
            ;2903 ;++
            ;2904 ; FUNCTIONAL DESCRIPTION:
            ;2905 ;
            ;2906 ; This routine is used to wait the specified number of machine cycles.
            ;2907 ; This routine is typically called to wait for an SBI write to
            ;2908 ; I/O space to complete.
            ;2909 ;
            ;2910 ; CALLING CONSTRAINT:
            ;2911 ;
            ;2912 ; =0
            ;2913 ;
            ;2914 ; INPUTS:
            ;2915 ;
            ;2916 ; d - Contains number of cycles to wait
            ;2917 ; alu.z condition code must not be set
            ;2918 ;
            ;2919 ; SIDE EFFECTS:
            ;2920 ;
            ;2921 ; d is destroyed
            ;2922 ;--
            ;2923 WAIT.LOOP:
U 11EA, 0018,0038,6180,0800,0010,108E  ;2924  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2925      ; ...is not set
            ;2926 =0
            ;2927 W.LOOP:
            ;2928  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 108E, 0819,0100,0580,0800,0010,108E  ;2929       j/W.LOOP
U 108F, 0000,003E,0180,0800,0000,0001  ;2930  return1    ; exit
            ;2931
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0561
ESKAR43.MCR                    MICRO2  1P(00)    29-JUL-85  15:01:16                   Page   9

            ;2932  .PAGE
            ;2933  .TOC "   RESET SUBTEST NUMBER"
            ;2934 ;**
            ;2935 ; FUNCTIONAL DESCRIPTION:
            ;2936 ;
            ;2937 ; Since some  of the tests will have to be restarted when two
            ;2938 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2939 ; the subtest counter to zero ( only for thoes tests that have
            ;2940 ; more than one subtest). This subroutine may also be necessary
            ;2941 ; when a test is being loop on.
            ;2942 ;
            ;2943 ; CALLING CONSTRAINT:
            ;2944 ;
            ;2945 ; =0
            ;2946 ; SIDE EFFECTS:
            ;2947 ;
            ;2948 ;
            ;2949 ;--
            ;2950 RESET.SUBTST:
            ;2951  RC[0F]_0,    ; Reset subtest number
U 11EB, 0003,003E,0180,09F8,0000,0001  ;2952   RETURN1    ; ...and return
            ;2953
```

```
            ;2954  .PAGE
            ;2955  .TOC "   WAIT REFRESH SUBROUTINE"
            ;2956 ;*****************************************************
            ;2957 ;++
            ;2958 ;
            ;2959 ; Functional Description:
            ;2960 ; ---------------------
            ;2961 ;
            ;2962 ; This subroutine provides the means of synchronizing the
            ;2963 ; diagnostic with the end of a refresh cycle on a MS780-E/H
            ;2964 ; memory.  It should be used by all the tests in which
            ;2965 ; timing is critical and delays due to refresh could
            ;2966 ; introduce false errors.
            ;2967 ;
            ;2968 ; Calling Constraint: =0
            ;2969 ; ------------------
            ;2970 ;
            ;2971 ;
            ;2972 ; Inputs:
            ;2973 ; ------
            ;2974 ;
            ;2975 ; RC[0E] must hold the I/O base address of the MS780-E/H
            ;2976 ;
            ;2977 ;
            ;2978 ; Outputs:
            ;2979 ; -------
            ;2980 ;
            ;2981 ; RETURN1 at the end of a refresh cycle.
            ;2982 ;
            ;2983 ;
            ;2984 ; Side Effects:
            ;2985 ; ------------
            ;2986 ;
            ;2987 ; The contents of the following registers will be lost:
            ;2988 ;
            ;2989 ; D, SC, CNFG Reg B
            ;2990 ;
            ;2991 ;--
            ;2992 ;*****************************************************
            ;2993 WAITREFRESH:
U 11EC, 0010,0038,0180,0970,0200,11ED  ;2994  VA_RC[0E]  ; Get address of CNFG Reg A
            ;2995  SC_K[.8],  ; Get data for CNFG  Reg B
U 11ED, 0000,003C,0180,0803,0084,71EE  ;2996   VA_VA+4  ; Point to CNFG Reg B
U 11EE, 0803,001C,0180,0800,0000,11EF  ;2997  D_NOT.MASK  ; Set bit (Refresh Lock Bit)
U 11EF, 0000,003C,0180,A800,0000,11F0  ;2998  CACHE.P_D[LONG]  ; Set bit 8 in CNFG Reg B
U 11F0, 0F00,003C,0180,0800,0000,11F1  ;2999  D_0   ; Get ready to clear the bit
U 11F1, 0000,003C,0180,A800,0000,11F2  ;3000  CACHE.P_D[LONG]  ; Clear bit 8 in CNFG Reg B
            ;3001 REFRESH.WAIT.LOOP:
            ;3002  D[LONG]_CACHE.P, ; Get contents of CNFG Reg B
U 11F2, 0000,003C,0180,C800,0084,71F3  ;3003   SC_K[.8] ; Point to bit 8
            ;3004  ALU_D.ANDNOT.MASK, ; Is bit 8 still set?
U 11F3, 0001,0024,0180,0800,0010,11F4  ;3005   CLK.UBCC ; ...
U 11F4, 0000,013C,0180,0800,0000,1090  ;3006  Z?   ;
U 1090, 0000,003C,0180,0800,0000,11F2  ;3007 =0 J/REFRESH.WAIT.LOOP ; Bit 8 is still set
U 1091, 0818,0038,1180,0800,0000,1092  ;3008  D_K[.4]   ; Number of cycles to wait for
```

```
            ;3009       ; ...refersh to end
U 1092, 0000.003D.0180.0800.0000.11EA  ;3010 =0 CALL[WAIT.LOOP]  ; Go and wait for refresh to end
U 1093, 0000.003E.0180.0800.0000.0001  ;3011  RETURN1   ; OK. Refresh should be done by now
            ;3012       ; ...Return to caller
            ;3013
            ;3014
            ;3015
            ;3016
```

```
;3017 .PAGE "TEST 19D CHECK SBI COMMANDS AND STALL USING SILO"
;3018 ;////////////////////////////////////////////////////////////////////////////
;3019 ;
;3020 ;
;3021 ;  SBI004.MIC
;3022 ;
;3023 ;============================================================================
;3024 ;++
;3025 ; TEST 1A0 SBI COMMANDS AND STALL USING SILO
;3026 ;
;3027 ;
;3028 ; TEST DESCRIPTION
;3029 ;
;3030 ;       THIS TEST CHECK THE COMMANDS ISSUED BY THE
;3031 ; CPU SBI INTERFACE.  THE STALL TIMING IS CHECKED
;3032 ; ALSO.  THE ADDRESS BIT #29 IS CHANGE TO SEE THAT
;3033 ; THE FUNCTIONS ARE PROPER IF A I/O REFERENCE IS MADE.
;3034 ; THE FOLLOWING TABLE IS USED TO DESCRIBE THE DIFFERENT CPU/
;3035 ; SBI OPERATIONS:
;3036 ;
;3037 ; NOTE: EXT R EXTENDED READ OPERATION
;3038 ;  EXT W EXTENDED WRITE OPERATION
;3039 ;  IR INTERLOCK READ
;3040 ;  IW INTERLOCK WRITE
;3041 ;  MASK R MASKED READ OPERATION
;3042 ;  MASK W MASKED WRITE OPERATION
;3043 ;
;3044 ;  IT SHOULD BE NOTED THAT READ.V.WCHK OPERATIONS ARE
;3045 ;  TRANSFORMED INTO INTERLOCK OPERATION IF I/O SPACE IS
;3046 ;  ADDRESSED.  THE IMPLICT INTERLOCK FLIP/FLOP (IMPLICT)
;3047 ;  IS USED TO TRANSFORM THE NEXT WRITE TO AN INTERLOCK WRITE
;3048 ;  OPERATION.
;3049 ;
;3050 ;
;3051 ; MCT/FUNCTION  NON I/O  I/O  COMMENT
;3052 ;    (ADDR #29=0) (ADDR #29=1)
;3053 ;
;3054 ;  LOCKWRITE.P  IW  IW
;3055 ;  LOCKWRITE.V.XCHK IW  IW
;3056 ;
;3057 ;  LOCKREAD.P  IR  IR
;3058 ;  LOCKREAD.V.NOCHK IR  IR
;3059 ;  LOCKREAD.V.WCHK  IR  IR
;3060 ;
;3061 ; EXTWRITE.P  EXT W  NOT ATTEMPTED
;3062 ;  NOTE:  IW(ANY) WILL CLEAR IMPLICT
;3063 ;
;3064 ; ================================================
;3065 ;
;3066 ; SUBTEST 1 GENERATE CONSTANTS FIND CPU ID
;3067 ;
;3068 ;
;3069 ; FUNCTION   OPERATION POINTER
;3070 ;    ^^^
;3071 ;
```

```
;3072 ; SUBTEST 2 LOCKWRITE.P (0) ADDR #29=0,1
;3073 ;  LOCKWRITE.V.XCHK(1) INTERLOCK WRITES
;3074 ;
;3075 ;
;3076 ;
;3077 ; SUBTEST 3 LOCKREAD.P (0) ADDR 29=0,1
;3078 ;  LOCKREAD.V.NOCHK(1) INTERLOCK READS
;3079 ;  LOCKREAD.V.WCHK (2)
;3080 ;
;3081 ;
;3082 ;
;3083 ; LOGIC DESCRIPTION
;3084 ;
;3085 ;    SILO LOGIC
;3086 ;    MCT DECODE ROMS
;3087 ;    IMPLIT FLIP/FLOP LOGIC
;3088 ;    MCT CONTROL LOGIC
;3089 ;
;3090 ;
;3091 ; ERROR LOGOUT
;3092 ;
;3093 ; DATA: I/O BASE ADDRESS OF MS780E/F CURRENTLY UNDER TEST
;3094 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER USED
;3095 ;  EXPECTED DATA FROM SILO
;3096 ;  RECEIVED DATA FROM SILO
;3097 ;  SILO COUNT, 0=1ST LOCATION AFTER LOCK
;3098 ;  OPERATION POINTER (OR ADDRESS FLAG)
;3099 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3100 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3101 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3102 ;  INTERRUPT FLAG:
;3103 ;   0=>NO INTERRUPT OCCURRED
;3104 ;   1=>AN INTERRUPT OCCURRED
;3105 ;  ADDRESS FLAG (0=NON I/O, 1=I/O ADDRESS)
;3106 ;
;3107 ; SEE SUBTEST FOR FULL DESCRIPTION
;3108 ;
;3109 ;--
;3110 ;================================================================================
;3111 ;    SCRATCH PADS
;3112 ; RA        II IIII = CPU I.D.
;3113 ; 0 C/A INTERLOCK WR  00II III0 1101 1100  0-0-0-0
;3114 ; 1 C/A INTERLOCK READ  00II III0 1101 0000  0-0-0-0
;3115 ; 2 C/A EXT READ   00II III0 1110 0000  0-0-0-0
;3116 ; 3 C/A MASKED READ   00II III0 1100 0100  0-0-0-0
;3117 ; 4 C/A MASKED WRITE  00II III0 1100 1000  0-0-0-0
;3118 ; 5 C/A EXTENDED WRITE  00II III0 1110 1100  0-0-0-0
;3119 ; 6 READ DATA   00II III0 0000 0000  0-0-0-0
;3120 ; 7 WRITE DATA (2ND FOR EXT W) 00II III1 0100 0000  0-0-0-0
;3121 ; 8 WRITE DATA (1ST FOR EXT W) 00II III1 0111 1100  0-0-0-0
;3122 ; 9 INTERLOCK MASK( SILO )  0100 0000 0000 0000  0-0-0-0
;3123 ; A CPU ARB (ARBITRITION)       0-0-0-0  <-TR->
;3124 ; B UNCONDITIONAL LOCK OFF 16 TIC   ^X 6000 0000
;3125 ; C SILO ID CONSTANT  00II III0 0000 0000  0-0-0-0
;3126 ; D CONFORMATION (ACK)   0-0-0  0001  0-0-0-0
```

```
            ;3127 ; E
            ;3128 ; F U-TRAP FLAG
            ;3129 ;
            ;3130 ;
            ;3131 ; RC
            ;3132 ; 0 CONFIG REG A ADDRESS    ^X 2000 2000
            ;3133 ; 1 CONFIG REG B ADDRESS    ^X 2000 2004
            ;3134 ; 2 CONFIG REG C ADDRESS    ^X 2000 2008
            ;3135 ; 3 CPU I.D.
            ;3136 ; 4 CPU ID SHIFTED TO MAINT ID POSITION
            ;3137 ; A ADDRESS FLAG
            ;3138 ; B OPERATION POINTER
            ;3139 ; C SILO COUNT
            ;3140 ; D DATA FROM SILO
            ;3141 ; E EXPECTED DATA
            ;3142 ; F # ARG TO CONSOLE
            ;3143 ;
            ;3144 ;================================================================================
            ;3145 ;
            ;3146 ;
            ;3147 ;
            ;3148 =0
U 1094, 0000,003D,0180,0800,0000,1127  ;3149 S05000: NEWTST
            ;3150  rc[0F]_K[.A],    ; ALL TR'S,MEMORIES AND MULTIPORTS
U 1095, 0F18,0038,F580,09F8,0000,1014  ;3151    D_0    ;
U 1014, 0000,003D,0180,0800,0000,1048  ;3152 =00 CALL[FIND.MS780E]   ; Find MS780-E/H on the SBI
U 1015, 0000,003D,1980,0800,0084,7038  ;3153  ERROR1[ZERO]    ; No MS780-E/H Found on the SBI
U 1016, 0000,003C,0180,0800,0000,11F5  ;3154  NOP
            ;3155 =
            ;3156 RESTART.TEST:
U 11F5, 001B,0000,8980,0800,0082,11F6  ;3157  SC_ALU,ALU_0-K[.D]   ; Get -13 in SC
U 11F6, 0810,0038,0180,0970,0000,11F7  ;3158  D_rc[0E]    ; Get Memory I/O Base Address
U 11F7, 0D00,003C,0180,0800,0000,11F8  ;3159  D_DAL.SC    ; Shift I/O base address to
            ;3160       ; ...extract TR level
U 11F8, 0819,0034,4980,0800,0000,11F9  ;3161  D_D.AND.K[.FF]    ; Mask unwanted bits and keep only
            ;3162       ; ...TR level in D Reg
U 11F9, 0000,003C,9180,3C00,0000,11FA  ;3163  ID[24]_D    ; Save in ID Reg 24
U 11FA, 0F00,003C,0180,0800,0000,11FB  ;3164  D_0
U 11FB, 0000,003C,7180,3C00,0000,11FC  ;3165  ID[COMP]_D    ; CLEAR SILO COMPARATOR
U 11FC, 0018,0038,0D80,09E8,0000,1018  ;3166  rc[0D]_K[.3]    ; Set flag for the fail-chain
U 1018, 0000,003D,0180,0800,0000,1004  ;3167 =00 CALL[SELECT.LOWER]   ; Try now to enable the Lower
            ;3168       ; ...Controller
U 1019, 0000,003C,0180,0800,0000,11FD  ;3169  J/TRY.UPPER    ; Lower ctrlr miscfg. try upper
U 101A, 0000,003C,0180,0800,0000,1096  ;3170  J/TEST.15    ; Lower ctrlr enabled, proceed
            ;3171       ; with test
            ;3172 =
            ;3173 TRY.UPPER:
U 11FD, 0018,0038,0D80,09E8,0000,1020  ;3174  RC[0D]_K[.3]    ; Set flag for fail-chain
U 1020, 0000,003D,0180,0800,0000,1010  ;3175 =00 CALL[SELECT.UPPER]   ; Try upper controller
```

```
            ;3176  .PAGE
            ;3177 ;
            ;3178 ;////////////////////////////////////////////////////////////////////////////////
            ;3179 ;
U 1021, 0000,003D,0980,0800,0084,7038  ;3180  ERROR1[.2]   ; Could NOT configure either
            ;3181       ; ...controller
            ;3182 ;
            ;3183 ;////////////////////////////////////////////////////////////////////////////////
            ;3184 ;
            ;3185 ; ERROR LOGOUT:
            ;3186 ;
            ;3187 ; DATA:
            ;3188 ;  I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;3189 ;  ERROR CODE ( CONTROLLERS MISCONFIGURED )
            ;3190 ;
            ;3191 ;
            ;3192 ;////////////////////////////////////////////////////////////////////////////////
            ;3193 ;
U 1022, 0000,003C,0180,0800,0000,1096  ;3194  J/TEST.15   ; Upper Controller is enabled
            ;3195       ; ...go execute the test
            ;3196 ;
            ;3197 ;********************************************************************************
            ;3198 ;////////////////////////////////////////////////////////////////////////////////
            ;3199 ;**
            ;3200 ;
            ;3201 ; THIS IS THE MAIN BODY OF THE TEST WHERE THE CONDITIONS
            ;3202 ; ARE BEING CHECKED OUT.
            ;3203 ;
            ;3204 ;--
            ;3205 ;////////////////////////////////////////////////////////////////////////////////
            ;3206 ;
            ;3207
            ;3208 ;
            ;3209 ;////////////////////////////////////////////////////////////////////////////////
            ;3210 ;**
            ;3211 ;
            ;3212 ; SUBTEST 1: CONSTANTS GENERATION
            ;3213 ;
            ;3214 ;--
            ;3215 ;////////////////////////////////////////////////////////////////////////////////
            ;3216 ;
            ;3217 =
            ;3218 =0
            ;3219 TEST.15:
U 1096, 0018,0039,8580,09F8,0000,1143  ;3220  CALL[SUBTEST1],rc[0F]_K[.C]
U 1097, 0F00,003C,ED80,0800,0084,71FE  ;3221  D_0,SC_K[.1B]    ;RA 9
U 11FE, 0000,003C,0D80,0800,0084,91FF  ;3222  SC_SC+K[.3]
U 11FF, 0801,001C,0180,0AC8,0000,1200  ;3223  R[9]_ALU,D_D.ORNOT.MASK
U 1200, 0000,003C,01E0,0800,0000,1201  ;3224  Q_D     ;RA B
U 1201, 0600,003C,0180,0800,0000,1202  ;3225  D_D.RIGHT
U 1202, 081D,0030,0180,0AD8,0000,1203  ;3226  R[0B]_ALU,D_D.OR.Q
            ;3227    ;============================================================
            ;3228    ; FIND CPU ID
            ;3229    ;============================================================
U 1203, 0003,003C,0180,0990,0000,1098  ;3230  RC[2]_0     ;SET INTIAL CPU ID
```

```
U 1098, 0000.003D.0180,0800.0000.113C  ;3231 =0 ERLOOP
U 1099, 0000.003C.0180,0800.0000.109A  ;3232 S05010: NOP
U 109A, 0000.003D.0180,0800.0000.11C8  ;3233 =0 CALL[INIT]
U 109B, 0F00.003C.7D80,0800.0084.7204  ;3234  SC_K[.18],D_0   ;GET FORCE MX BIT
U 1204, 0000.003C.1180,0800.0084.9205  ;3235  SC_SC+K[.4]
U 1205, 0801.001C.0180,0800.0000.1206  ;3236  D_D.ORNOT.MASK
U 1206, 0000.003C.75F0,2C00.0000.1207  ;3237  Q_ID[MAINT]   ;GET CACHE OFF BITS
U 1207, 081D.0030.0180,0800.0000.1208  ;3238  D_D[OR]Q   ;OR THE TWO TOGETHER
U 1208, 0000.003C.7580,3C00.0000.1209  ;3239  ID[MAINT]_D   ;SAVE IT IN DESTINATION
U 1209, 0810.0038.0180,0910.0000.120A  ;3240  D_RC[2]
U 120A, 0019.0020.6580,0800.0010.120B  ;3241  ALU_D[XOR]K[.10],CLK.UBCC
U 120B, 0819.0114.0580,0990.0000.109C  ;3242  Z?,RC[2]_ALU,D_D+K[.1]  ;GET NEXT ID
U 109C, 0000.003C.0180,0800.0000.120C  ;3243 =0 J/S05020
U 109D, 0000.003D.0180,0800.0000.113D  ;3244  ERROR1    ;DIDN'T SEE A MULTIPLE XMITTER
        ;3245       ;AFTER ALL 16 ID'S CHECKED
U 120C, 0000.003C.21F8,0800.0084.720D  ;3246 S05020: SC_K[.14],Q_0   ;SHIFT TEST ID TO MAINT
U 120D, 0000.003C.0D80,0800.0084.920E  ;3247  SC_SC+K[.3]   ; ID POSITION
U 120E, 0D00.003C.0180,0800.0000.120F  ;3248  D_DAL.SC
U 120F, 0000.003C.75F0,2C00.0000.1210  ;3249  Q_ID[MAINT]
U 1210, 081D.0030.0180,0800.0000.1211  ;3250  D_D[OR]Q   ;GOT FORCE MX,ID AND
U 1211, 0000.003C.7580,3C00.0000.1212  ;3251  ID[MAINT]_D   ;  CACHE OFF
U 1212, 0018.0038.1980,0800.0200.1213  ;3252  VA_K[ZERO]
U 1213, 0000.003C.0180,A800.0000.109E  ;3253  MCT/WRITE.P   ;WRITE CAUSING MULTIPLE
        ;3254       ; ...XMITTER FAULT
        ;3255 =0 D_K[.4].    ; WAIT 6 CLOCK TICKS
U 109E, 0818.0039.1180,0800.0000.11EA  ;3256   CALL[WAIT.LOOP]  ; ...
U 109F, 0F00.003C.0180,0800.0000.1214  ;3257  D_0
U 1214, 0000.003C.7580,3C00.0000.1024  ;3258  ID[MAINT]_D   ;CLEAR FORCE FAULT
U 1024, 0000.003D.0180,0800.0000.11CB  ;3259 =00 CALL[SBIFLT]
U 1025, 0000.003C.0180,0800.0000.121E  ;3260  J/S05030   ;NO SBI FAULT,GOT ID
U 1026, 0818.0038.85F8,0800.0000.1027  ;3261  D_K[.C],Q_0   ;CHECK THE FAULT BITS SET
U 1027, 0000.003C.7D80,0800.0084.7215  ;3262  SC_K[.18]
U 1215, 0D00.003C.0180,0800.0000.1216  ;3263  D_DAL.SC
U 1216, 0001.003C.0180,09E0.0000.1217  ;3264  RC[0C]_D   ;SAVE EXPECTED DATA
U 1217, 0818.0038.F1F8,0800.0000.1218  ;3265  D_K[.FFFC],Q_0   ;GENERATE MASK FOR SBI FAULT BITS
U 1218, 0000.003C.7D80,0800.0084.7219  ;3266  SC_K[.18]
U 1219, 0D00.003C.0180,0800.0000.121A  ;3267  D_DAL.SC
U 121A, 0000.003C.6DF0,2C00.0000.121B  ;3268  Q_ID[FAULT]   ;GET FAULT REGISTER IMAGE
U 121B, 081D.0034.0180,09D8.0000.121C  ;3269  RC[0B]_ALU,D_ALU,ALU_D[AND]Q ;MASK AND SAVE
U 121C, 0011.0020.0180,0960.0010.121D  ;3270  ALU_D[XOR]RC[0C],CLK.UBCC ;ARE THE RIGHT BITS SET
U 121D, 0000.013C.0180,0800.0000.10A0  ;3271  Z?
U 10A0, 0000.003D.0180,0800.0000.113D  ;3272 =0 ERROR1    ;WRONG FAULT BITS SET
U 10A1, 0000.003C.0180,0800.0000.1099  ;3273  J/S05010   ;GO TRY NEXT ID
U 121E, 0810.0038.01F8,0910.0000.121F  ;3274 S05030: D_RC[2],Q_0   ;SHIFT ID TO SBI SILO ID
U 121F, 0000.003C.7D80,0800.0084.7220  ;3275  SC_K[.18]   ; POSITION
U 1220, 0000.003C.0580,0800.0084.9221  ;3276  SC_SC+K[.1]
U 1221, 0D00.003C.0180,0800.0000.1222  ;3277  D_DAL.SC
U 1222, 0001.003C.0180,0998.0000.1223  ;3278  RC[3]_D   ;SAVE IT IN RC 4
U 1223, 0818.0038.89F8,0800.0000.1224  ;3279  D_K[.D],Q_0   ;RA 0
U 1224, 0000.003C.1180,0800.0084.7225  ;3280  SC_K[.4]
U 1225, 0D00.003C.0180,0800.0000.1226  ;3281  D_DAL.SC
U 1226, 0819.0030.8580,0800.0000.1227  ;3282  D_D.OR.K[.C]
U 1227, 0000.003C.6580,0800.0084.7228  ;3283  SC_K[.10]
U 1228, 0D00.003C.0180,0800.0000.1229  ;3284  D_DAL.SC
U 1229, 0811.0030.0180,0918.0000.122A  ;3285  D_ALU,ALU_D[OR]RC[3]
```

```
U 122A. 0001.003C.0180.0A80.0000.122B  ;3286  R[0]_D
U 122B. 0818.0038.89F8.0800.0000.122C  ;3287  D_K[.D],Q_0    ;RA 1
U 122C. 0000.003C.1180.0800.0084.922D  ;3288  SC_SC+K[.4]
U 122D. 0D00.003C.0180.0800.0000.122E  ;3289  D_DAL.SC
U 122E. 0811.0030.0180.0918.0000.122F  ;3290  D_ALU,ALU_D[OR]RC[3]
U 122F. 0F01.003C.0180.0A88.0000.10A2  ;3291  R[1]_D,D_0
U 10A2. 0000.003D.0180.0800.0000.115B  ;3292 =0 CALL[DCE]    ;RA 2
U 10A3. 0000.003C.6580.0800.0084.7230  ;3293  SC_K[.10]
U 1230. 0000.003C.1180.0800.0084.9231  ;3294  SC_SC+K[.4]
U 1231. 0D00.003C.0180.0800.0000.1232  ;3295  D_DAL.SC
U 1232. 0811.0030.0180.0918.0000.1233  ;3296  D_ALU,ALU_D[OR]RC[3]
U 1233. 0001.003C.0180.0A90.0000.1234  ;3297  R[2]_D
U 1234. 0818.0038.85F8.0800.0000.1235  ;3298  D_K[.C],Q_0    ;RA 3
U 1235. 0000.003C.1180.0800.0084.7236  ;3299  SC_K[.4]
U 1236. 0D00.003C.0180.0800.0000.1237  ;3300  D_DAL.SC
U 1237. 0819.0030.1180.0800.0000.1238  ;3301  D_D.OR.K[.4]
U 1238. 0000.003C.6580.0800.0084.7239  ;3302  SC_K[.10]
U 1239. 0D00.003C.0180.0800.0000.123A  ;3303  D_DAL.SC
U 123A. 0811.0030.0180.0918.0000.123B  ;3304  D_ALU,ALU_D[OR]RC[3]
U 123B. 0001.003C.0180.0A98.0000.123C  ;3305  R[3]_D
U 123C. 0818.0038.85F8.0800.0000.123D  ;3306  D_K[.C],Q_0    ;RA 4
U 123D. 0000.003C.1180.0800.0084.723E  ;3307  SC_K[.4]
U 123E. 0D00.003C.0180.0800.0000.123F  ;3308  D_DAL.SC
U 123F. 0819.0030.0180.0800.0000.1240  ;3309  D_D.OR.K[.8]
U 1240. 0000.003C.6580.0800.0084.7241  ;3310  SC_K[.10]
U 1241. 0D00.003C.0180.0800.0000.1242  ;3311  D_DAL.SC
U 1242. 0811.0030.0180.0918.0000.1243  ;3312  D_ALU,ALU_D[OR]RC[3]
U 1243. 0001.003C.0180.0AA0.0000.1244  ;3313  R[4]_D
U 1244. 0811.0030.0180.0918.0000.1245  ;3314  D_ALU,ALU_D[OR]RC[3]  ;RA 5
U 1245. 0000.003C.2180.0800.0084.7246  ;3315  SC_K[.14]
U 1246. 0000.003C.0580.0800.0084.9247  ;3316  SC_SC+K[.1]    ;SET BIT #21
U 1247. 0801.001C.0180.0AA8.0000.1248  ;3317  R[5]_ALU,D_D.ORNOT.MASK
U 1248. 0810.0038.0180.0918.0000.1249  ;3318  D_RC[3]     ;RA 6
U 1249. 0001.003C.0180.0AB0.0000.124A  ;3319  R[6]_D
U 124A. 0818.0038.F5F8.0800.0000.124B  ;3320  D_K[.A],Q_0    ;RA 7
U 124B. 0000.003C.2180.0800.0084.724C  ;3321  SC_K[.14]
U 124C. 0000.003C.0580.0800.0084.924D  ;3322  SC_SC+K[.1]    ;SKIP 21 PLACES FOR LEFT
U 124D. 0D00.003C.0180.0800.0000.124E  ;3323  D_DAL.SC
U 124E. 0811.0030.0180.0918.0000.124F  ;3324  D_ALU,ALU_D[OR]RC[3]
U 124F. 0001.003C.6580.0AB8.0084.7250  ;3325  R[7]_D,SC_K[.10]
U 1250. 0818.0038.61F8.0800.0000.1251  ;3326  D_K[.F],Q_0    ;RA 8
U 1251. 0000.003C.0980.0800.0084.9252  ;3327  SC_SC+K[.2]
U 1252. 0D00.003C.0180.0800.0000.1253  ;3328  D_DAL.SC
U 1253. 080D.0030.0180.0A38.0000.1254  ;3329  D_ALU,ALU_D[OR]R[7]
U 1254. 0001.003C.0180.0AC0.0000.1255  ;3330  R[8]_D
U 1255. 0F10.0038.0180.0910.0082.1256  ;3331  SC_ALU,ALU_RC[2],D_0  ;RA A
U 1256. 0801.001C.0180.0AD0.0000.1257  ;3332  R[0A]_ALU,D_D.ORNOT.MASK
U 1257. 0818.0038.4580.0800.0000.1258  ;3333  D_K[.8000]    ;RA D
U 1258. 0500.003C.0180.0800.0000.1259  ;3334  D_D.LEFT
U 1259. 0001.003C.0180.0AE8.0000.10A4  ;3335  R[0D]_D
            ;3336 ;
            ;3337 ;================================================================================
            ;3338 ;*
            ;3339 ;SUBTEST 2   INTERLOCK WRITES
            ;3340 ;
```

```
            ;3341 ;  THIS TEST CHECK THE INTERLOCK WRITE FUNCTION
            ;3342 ;  ON THE SBI USING THE SILO.  THE INTERLOCK
            ;3343 ;  WRITES WILL BE CHECK BOTH IN MEMORY SPACE
            ;3344 ;  (ADDR #29=0) AND I/O SPACE (ADDR #29=1)
            ;3345 ;
            ;3346 ;  FUNCTION OPERATION POINTER
            ;3347 ;  -------- ------------------------------------
            ;3348 ;  MCT/LOCKWRITE.P  (0) ADDR #29=0,1
            ;3349 ;  MCT/LOCKWRITE.V.XCHK (1) INTERLOCK WRITES
            ;3350 ;
            ;3351 ;  PRECEED INTERLOCK WRITE WITH AN INTERLOCK
            ;3352 ;  READ TO PREVENT INTERLOCK SEQUENCE FAULT.
            ;3353 ;
            ;3354 ;  SILO CONTENTS:
            ;3355 ;
            ;3356 ;   *NULL (4 TICS)
            ;3357 ;   *CPU ARB (IF I.D. NOT = 10)
            ;3358 ;   *C/A INTERLOCK WRITE, AND SBI HOLD
            ;3359 ;    *WRITE DATA
            ;3360 ;    ACK
            ;3361 ;    ACK
            ;3362 ;    NULL (UNTIL COUNT=F)
            ;3363 ;
            ;3364 ; NOTE:  *  CYCLES WILL HAVE SBI INTERLOCK SET IF
            ;3365 ;      NON-I/O REFERENCE IS UNDER TEST.  MEMORY
            ;3366 ;      DOES NOT HOLD SBI INTERLOCK LINE ON
            ;3367 ;      I/O REFERENCES.
            ;3368 ;
            ;3369 ;
            ;3370 ;ERROR LOGOUT
            ;3371 ;  DATA: EXPECTED DATA
            ;3372 ;   DATA FROM SILO
            ;3373 ;   SILO COUNT, 0=1ST LOCATION AFTERLOCK
            ;3374 ;   OPERATION POINTER (0-1)
            ;3375 ;   ADDRESS FLAG (0=>NON I/O, 1=>I/O REFERENCE)
            ;3376 ;   TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
            ;3377 ;   ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;3378 ;   ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;3379 ;   ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;3380 ;   INTERRUPT FLAG:
            ;3381 ;    0=>NO INTERRUPT OCCURRED
            ;3382 ;    1=>AN INTERRUPT OCCURRED
            ;3383 ;
            ;3384 ;-
            ;3385 ;================================================================================
            ;3386 ;
U 10A4, 0000,003D,0180,0800,0000,1141  ;3387 =0 SUBTEST
U 10A5, 0003,003C,0180,09C0,0000,125A  ;3388  rc[8]_0    ;CLEAR ADDRESS FLAG
U 125A, 0003,003C,0180,09C8,0000,10A6  ;3389 S05300: RC[9]_0    ;SET OPERATION POINTER
U 10A6, 0000,003D,0180,0800,0000,113C  ;3390 =0 ERLOOP
            ;3391 RETRY3:
U 10A7, 0000,003C,0180,0800,0000,10A8  ;3392 S05310: NOP
U 10A8, 0000,003D,0180,0800,0000,11C8  ;3393 =0 CALL[INIT]
U 10A9, 0010,0038,0180,0940,0010,125B  ;3394  ALU_rc[8],CLK.UBCC
U 125B, 0F18,0138,1980,0800,0200,10AA  ;3395  Z?,VA_K[ZERO],D_0
```

```
U 10AA, 0F10,0038,0180,0970,0200,10AB  ;3396 =0 VA_rc[0E],D_0   ;I/O REFERENCE
U 10AB, 0000,003C,0180,E800,0000,10AC  ;3397  MCT/LOCKREAD.P   ;INTERLOCK READ TO
          ;3398       ; SET MEMORY INTERLOCK
          ;3399       ; FLOP
          ;3400 =0 D_K[.8],
U 10AC, 0818,0039,0180,0800,0000,11EA  ;3401   CALL[WAIT.LOOP]
          ;3402 ;=00 CALL[NOINTR]
          ;3403 ; ERROR1    ;UNEXPECTED INTERRUPT
U 10AD, 0800,003C,0180,0A58,0000,125C  ;3404  D_R[0B]    ;ENABLE SILO
U 125C, 0810,0038,7180,3D48,0000,125D  ;3405  ID[COMP]_D,D_RC[9]
U 125D, 0000,193C,0180,0800,0000,101E  ;3406  DO?    ;BRANCH ON OPERATION POINTER
U 101E, 0000,003C,0180,B800,0000,10AE  ;3407 =1110 MCT/LOCKWRITE.P,J/S05320 ; (0)
U 101F, 0000,003C,0180,3800,0000,10AE  ;3408  MCT/LOCKWRITE.V.XCHK,J/S05320 ; (1)
          ;3409 =0
          ;3410 S05320: D_K[.18],
U 10AE, 0818,0039,7D80,0800,0000,11EA  ;3411   CALL[WAIT.LOOP]   ;LET SILO FILL, 24 TICS
U 10AF, 0000,003C,0180,0800,0000,1028  ;3412  NOP
U 1028, 0000,003D,0180,0800,0000,11D1  ;3413 =00 CALL[SILINT]   ;CHECK SILO INTERRUPT
U 1029, 0000,003D,0180,0800,0000,113D  ;3414  ERROR1    ;DIDN'T GET SILO INTERRUPT
          ;3415   ;=================================================================
          ;3416   ;
          ;3417   ; CHECK SILO CONTENTS
          ;3418   ;
          ;3419   ;=================================================================
U 102A, 0003,003C,0180,09E0,0000,125E  ;3420  RC[0C]_0   ;SET EXPECTED DATA
U 125E, 0003,003C,0180,09D0,0000,10B0  ;3421 = RC[0A]_0   ;SET INITIAL COUNT
U 10B0, 0000,003D,F580,0800,0084,7145  ;3422 =0 SETRCF[.A]
          ;3423   ;=================================================================
          ;3424   ;SEE NULL (IF NON I/O, SEE INTERLOCK SET) 4 LOCATIONS
          ;3425   ;=================================================================
U 10B1, 0010,0038,0180,0940,0010,125F  ;3426 S05330: ALU_rc[8],CLK.UBCC
          ;3427 ;
          ;3428 ;-------------------------------------------------------------------------
          ;3429 ;
          ;3430 ; TEMOPORARILY REMOVED AND REPLACED WITH "Z?,D_0"
          ;3431 ; TESTING PURPOSES UNTIL ESD ADDS INTLK SIGNAL TO SBI
          ;3432 ;
          ;3433 ; VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
          ;3434 ; Z?,D_0 ; Mod for D_R[9]   ;NON-I/O SEE INTERLOCK
          ;3435 ; ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
          ;3436 ;
          ;3437 ;-------------------------------------------------------------------------
          ;3438 ;
U 125F, 0F00,013C,0180,0800,0000,10B2  ;3439  Z?,D_0
U 10B2, 0F00,003C,0180,0800,0000,10B3  ;3440 =0 D_0    ;I/O SPACE, SEE NULL
U 10B3, 0001,003C,61F0,2DE0,0000,1260  ;3441  RC[0C]_D,Q_ID[SILO]
U 1260, 0001,203C,0180,09D8,0000,1261  ;3442  RC[0B]_Q
U 1261, 001D,0020,0180,0800,0010,1262  ;3443  ALU_D[XOR]Q,CLK.UBCC
U 1262, 0810,0138,0180,0950,0000,10B4  ;3444  Z?,D_RC[0A]
U 10B4, 0000,003D,0180,0800,0000,113D  ;3445 =0 ERROR1    ;SILO NULL NOT CORRECT
U 10B5, 0019,0020,0D80,0800,0010,1263  ;3446  ALU_D[XOR]K[.3],CLK.UBCC
U 1263, 0819,0114,0580,09D0,0000,10B6  ;3447  Z?,RC[0A]_ALU,D_D+K[.1]
U 10B6, 0000,003C,0180,0800,0000,10B1  ;3448 =0 J/S05330
U 10B7, 0818,0038,6580,0800,0000,1264  ;3449  D_K[.10]   ;CPU I.D.=10?
U 1264, 0011,0020,0180,0910,0010,1265  ;3450  ALU_D[XOR]RC[2],CLK.UBCC
```

```
ZZ-ESKAR-V22  TEST 19D CHECK SBI COMMANDS AND STALL USING SILO                                  SEQ 0572
ESKAR43.MCR                    MICRO2  1P(00)    29-JUL-85  15:01:16                          Page  10

U 1265. 0810.0138,0180.0950.0000.10B8  ;3451  Z?,D_RC[0A]
U 10B8. 0000.003C,0180.0800.0000.1266  ;3452 =0 J/S05340    ;ID NOT 10, CHECK ARB
U 10B9. 0000.003C,0180.0800.0000.126C  ;3453  J/S05350    ;ID = 10, CHECK C/A
            ;3454    ;=============================================================
            ;3455    ;SEE CPU ARB(NON-I/O SEE INTERLOCK SET)
            ;3456    ;=============================================================
U 1266. 0010.0038,0180.0940.0010.1267  ;3457 S05340: ALU_rc[8],CLK.UBCC
U 1267. 0F00.013C,0180.0800.0000.10BA  ;3458  Z?,D_0 ; Mod for D_R[9]    ;NON I/O SPACE
U 10BA. 0F00.003C,0180.0800.0000.10BB  ;3459 =0 D_0     ;I/O SPACE
U 10BB. 080D.0030,0180.0A50.0000.1268  ;3460  D_ALU,ALU_D[OR]R[0A]
U 1268. 0001.003C,61F0.2DE0.0000.1269  ;3461  RC[0C]_D,Q_ID[SILO]
U 1269. 0001.203C,0180.09D8.0000.126A  ;3462  RC[0B]_Q
U 126A. 001D.0020,0180.0800.0010.126B  ;3463  ALU_D[XOR]Q,CLK.UBCC
U 126B. 0810.0138,0180.0950.0000.10BC  ;3464  Z?,D_RC[0A]
U 10BC. 0000.003D,0180.0800.0000.113D  ;3465 =0 ERROR1     ;DIDN'T SEE ARB CORRECTLY
            ;3466    ;=============================================================
            ;3467    ;CHECK C/A, INTERLOCK WRITE,HOLD
            ;3468    ; (NON-I/O SEE SBI INTERLOCK SET)
            ;3469    ;=============================================================
U 10BD. 0819.0014,0580.09D0.0000.126C  ;3470  RC[0A]_ALU,D_D+K[.1]
U 126C. 0010.0038,0180.0940.0010.126D  ;3471 S05350: ALU_rc[8],CLK.UBCC  ;I/O FLAG SET?
U 126D. 0F00.013C,0180.0800.0000.10BE  ;3472  Z?,D_0 ; Mod for D_R[9]    ;NON-I/O
U 10BE. 0F00.003C,0180.0800.0000.10BF  ;3473 =0 D_0     ;I/O SPACE
U 10BF. 080D.0030,0180.0A00.0000.126E  ;3474  D_ALU,ALU_D[OR]R[0]
U 126E. 0819.0030,0580.0800.0000.126F  ;3475  D_ALU,ALU_D[OR]K[.1]
U 126F. 0001.003C,61F0.2DE0.0000.1270  ;3476  RC[0C]_D,Q_ID[SILO]
U 1270. 0001.203C,0180.09D8.0000.1271  ;3477  RC[0B]_Q
U 1271. 001D.0020,0180.0800.0010.1272  ;3478  ALU_D[XOR]Q,CLK.UBCC
U 1272. 0810.0138,0180.0950.0000.10C0  ;3479  Z?,D_RC[0A]
U 10C0. 0000.003D,0180.0800.0000.113D  ;3480 =0 ERROR1     ;BAD COMMAND/ADDRESS
            ;3481    ;=============================================================
            ;3482    ;CHECK FOR WRITE DATA(INTERLOCK SET)
            ;3483    ;=============================================================
U 10C1. 0819.0014,0580.09D0.0000.1273  ;3484  RC[0A]_ALU,D_D+K[.1]
U 1273. 0010.0038,0180.0940.0010.1274  ;3485  ALU_rc[8],CLK.UBCC  ;I/O REFERENCE ???
U 1274. 0F00.013C,0180.0800.0000.10C2  ;3486  Z?,D_0 ; Mod for D_R[9]    ;NON I/O REFERENCE
U 10C2. 0F00.003C,0180.0800.0000.10C3  ;3487 =0 D_0     ;I/O REFERENCE
U 10C3. 080D.0030,0180.0A40.0000.1275  ;3488  D_ALU,ALU_D[OR]R[8]  ;SET INTERLOCK BIT
U 1275. 0001.003C,61F0.2DE0.0000.1276  ;3489  RC[0C]_D,Q_ID[SILO]
U 1276. 0001.203C,0180.09D8.0000.1277  ;3490  RC[0B]_Q
U 1277. 001D.0020,0180.0800.0010.1278  ;3491  ALU_D[XOR]Q,CLK.UBCC
U 1278. 0810.0138,1980.0950.0084.70C4  ;3492  Z?,D_RC[0A],SC_K[ZERO]
U 10C4. 0000.003D,0180.0800.0000.113D  ;3493 =0 ERROR1     ;BAD WRITE DATA IN SILO
            ;3494    ;=============================================================
            ;3495    ;SEE 2 ACKS FROM MEMORY
            ;3496    ;=============================================================
U 10C5. 0819.0014,0580.09D0.0084.9279  ;3497 S05360: RC[0A]_ALU,D_D+K[.1],SC_SC+K[.1]
U 1279. 0018.0038,1DC0.0800.0000.127A  ;3498  Q_ALU,ALU_SC
U 127A. 0019.2020,0580.0800.0010.127B  ;3499  ALU_Q,XOR.K[.1],CLK.UBCC ;1ST ACK ??
U 127B. 0F00.013C,0180.0800.0000.10C6  ;3500  Z?,D_0 ; Mod for D_R[9]    ;IF 1ST ACK, SEE INTERLOCK SET
U 10C6. 0F00.003C,0180.0800.0000.10C9  ;3501 =0 D_0,J/S05365    ;2ND ACK ,DON'T SEE INTERLOCK
U 10C7. 0010.0038,0180.0940.0010.127C  ;3502  ALU_rc[8],CLK.UBCC  ;I/O REFERENCE ???
U 127C. 0000.013C,0180.0800.0000.10C8  ;3503  Z?
U 10C8. 0F00.003C,0180.0800.0000.10C9  ;3504 =0 D_0     ;I/O REFERENCE
U 10C9. 080D.0030,61F0.2E68.0000.127D  ;3505 S05365: D_ALU,ALU_D[OR]R[0D],Q_ID[SILO]
```

```
U 127D. 0001.003C.0180.09E0.0000.127E  ;3506  RC[0C]_D
U 127E. 0001.203C.0180.09D8.0000.127F  ;3507  RC[0B]_Q
U 127F. 001D.0020.0180.0800.0010.1280  ;3508  ALU_D[XOR]Q,CLK.UBCC
U 1280. 0810.0138.0180.0950.0000.10CA  ;3509  Z?,D_RC[0A]
U 10CA. 0000.003D.0180.0800.0000.113D  ;3510 =0 ERROR1     ;NOT AN ACK IN SILO
U 10CB. 0018.0038.1DC0.0800.0000.1281  ;3511  Q_ALU,ALU_SC    ;SC=2???
U 1281. 0019.2020.0980.0800.0010.1282  ;3512  ALU_Q,XOR.K[.2],CLK.UBCC
U 1282. 0810.0138.0180.0950.0000.10CC  ;3513  Z?,D_RC[0A]
U 10CC. 0000.003C.0180.0800.0000.10C5  ;3514 =0 J/S05360
          ;3515    ;=============================================================
          ;3516    ;SEE NULL CYCLES
          ;3517    ;=============================================================
U 10CD. 0819.0014.0580.09D0.0000.1283  ;3518 S05370: RC[0A]_ALU,D_D+K[.1]
U 1283. 0003.003C.61F0.2DE0.0000.1284  ;3519  RC[0C]_0,Q_ID[SILO]
U 1284. 0001.203C.0180.09D8.0010.1285  ;3520  RC[0B]_Q,CLK.UBCC
U 1285. 0000.013C.0180.0800.0000.10CE  ;3521  Z?
U 10CE. 0000.003D.0180.0800.0000.113D  ;3522 =0 ERROR1     ;NOT A NULL CYCLE
U 10CF. 0019.0020.6180.0800.0010.1286  ;3523  ALU_D[XOR]K[.F],CLK.UBCC
U 1286. 0810.0138.0180.0950.0000.10D0  ;3524  Z?,D_RC[0A]
U 10D0. 0000.003C.0180.0800.0000.10CD  ;3525 =0 J/S05370    ;GO REPEAT LOOP
          ;3526    ;=============================================================
          ;3527    ;
          ;3528    ;CHECK OPERATION POINTER AND ADDRESS FLAG
          ;3529    ;
          ;3530    ;=============================================================
U 10D1. 0810.0038.0180.0948.0000.1287  ;3531  D_RC[9]    ;OPERATION POINTER =2
U 1287. 0019.0020.0980.0800.0010.1288  ;3532  ALU_D[XOR]K[.2],CLK.UBCC
U 1288. 0819.0114.0580.09C8.0000.10D2  ;3533  Z?,RC[9]_ALU,D_D+K[.1]
U 10D2. 0000.003C.0180.0800.0000.10A7  ;3534 =0 J/S05310    ;REPEAT LOOP NEW OPERATION
U 10D3. 0010.0038.0180.0940.0010.1289  ;3535  ALU_rc[8],CLK.UBCC  ;=2,CHECK ADDRESS FLAG
U 1289. 0000.013C.0180.0800.0000.10D4  ;3536  Z?
U 10D4. 0000.003C.0180.0800.0000.128A  ;3537 =0 J/S05380    ;FLAG SET END TEST
U 10D5. 0018.0038.0580.09C0.0000.125A  ;3538  rc[8]_K[.1],J/S05300  ;REPEAT LOOP, WITH
          ;3539         ; NEW ADDRESS
U 128A. 0000.003C.0180.0800.0000.10D6  ;3540 S05380: NOP     ;END SUBTEST
          ;3541 ;
          ;3542 ;=============================================================================
          ;3543 ;*
          ;3544 ;SUBTEST 3  INTERLOCK READS
          ;3545 ;
          ;3546 ;  THIS TEST CHECKS THE INTERLOCK READ FUNCTION
          ;3547 ;  ON THE SBI USING THE SILO.  THE INTERLOCK
          ;3548 ;  READ WILL BE CHECKED IN MEMORY AND I.O SPACE
          ;3549 ;  (ADDRESS BIT #29 = 1,0).  THE FOLLOWING FUNCTION WILL
          ;3550 ;  BE CHECKED:
          ;3551 ;
          ;3552 ;  FUNCTION OPERATION POINTER
          ;3553 ;  -------- ------------------------------------
          ;3554 ;  LOCKREAD.P  (0) SEE INTERLOCK READ
          ;3555 ;  LOCKREAD.V.NOCHK (1) ADDRS #29 = 0,1
          ;3556 ;  LOCKREAD.V.WCHK  (2)
          ;3557 ;
          ;3558 ;  A 104 U-SEC WAIT AFTER EACH OPERATION WILL BE
          ;3559 ;  ALLOWED TO LET THE INTERLOCK TIMER IN MEMORY
          ;3560 ;  TO CLEAR.
```

```
                ;3561 ;
                ;3562 ;  SILO CONTENTS:
                ;3563 ;
                ;3564 ;   NULL (4 TICS)
                ;3565 ;   CPU ARB (IF I.D. NOT = 10)
                ;3566 ;   C/A INTERLOCK READ
                ;3567 ;   NULL(CPU DRIVES INTERLOCK LINE)
                ;3568 ;  *ACK(MEMORY DRIVES INTERLOCK, STARTING HERE)
                ;3569 ;  *MEMORY ARB
                ;3570 ;  *READ DATA
                ;3571 ;  *ACK
                ;3572 ;  *NULL( UNTILL COUNT = F )
                ;3573 ;
                ;3574 ; NOTE:  *  THESE CYCLE WILL HAVE THE INTERLOCK BIT SET ONLY
                ;3575 ;     IF NON I/O REFERENCE IS MADE.  MEMORY DOES NOT
                ;3576 ;     HOLD INTERLOCK LINE ON INTERLOCK I/O OPERATIONS.
                ;3577 ;
                ;3578 ;ERROR LOGOUT
                ;3579 ;
                ;3580 ; DATA: EXPECTED DATA
                ;3581 ;  DATA FROM SILO
                ;3582 ;  SILO COUNT, 0=1ST LOCATION AFTER LOCK
                ;3583 ;  OPERATION POINTER (0-2)
                ;3584 ;  ADDRESS FLAG (0=>ADDR #29=0, 1=>ADDR #29=1)
                ;3585 ;  TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
                ;3586 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
                ;3587 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
                ;3588 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;3589 ;  INTERRUPT FLAG:
                ;3590 ;   0=>NO INTERRUPT OCCURRED
                ;3591 ;   1=>AN INTERRUPT OCCURRED
                ;3592 ;
                ;3593 ;-
                ;3594 ;================================================================================
                ;3595 ;
                ;3596 ;
U 10D6, 0000,003D,0180,0800,0000,1141   ;3597 =0 SUBTEST
U 10D7, 0003,003C,0180,09C0,0000,128B   ;3598  rc[8]_0    ;RESET ADDRESS FLAG
U 128B, 0003,003C,0180,09C8,0000,10D8   ;3599 S05400: RC[9]_0    ;RESET OPERATION POINTER
U 10D8, 0000,003D,0180,0800,0000,113C   ;3600 =0 ERLOOP
                ;3601 RETRY4:
U 10D9, 0000,003C,0180,0800,0000,10DA   ;3602 S05410: NOP
U 10DA, 0000,003D,0180,0800,0000,11C8   ;3603 =0 CALL[INIT]
U 10DB, 0000,003C,0180,0800,0000,10DC   ;3604  NOP
U 10DC, 0000,003D,0180,0800,0000,11EC   ;3605 =0 CALL[WAITREFRESH]
U 10DD, 0010,0038,0180,0940,0010,128C   ;3606  ALU_rc[8],CLK.UBCC   ;SET ADDRESS
U 128C, 0018,0138,1980,0800,0200,10DE   ;3607  Z?,VA_K[ZERO]    ;NON I/O REFERENCE
U 10DE, 0010,0038,0180,0970,0200,10DF   ;3608 =0 VA_RC[0E]    ;I/O REFERENCE
U 10DF, 0000,003C,0180,0800,0000,10E0   ;3609  NOP
U 10E0, 0800,003D,F580,0A58,0084,7145   ;3610 =0 D_R[0B],SETRCF[.A]
U 10E1, 0810,0038,7180,3D48,0000,128D   ;3611  ID[COMP]_D,D_RC[9]   ;ENABLE SILO, SET OPERATION
U 128D, 0000,193C,0180,0800,0000,102C   ;3612  D1-0?     ;BRANCH TO OPERATION
U 102C, 0000,003C,0180,E800,0000,10E2   ;3613 =1100 MCT/LOCKREAD.P,J/S05420   ; (0)
U 102D, 0000,003C,0180,6800,0000,10E2   ;3614  MCT/LOCKREAD.V.NOCHK,J/S05420 ; (1)
U 102E, 0000,003C,0180,7000,0000,10E2   ;3615  MCT/LOCKREAD.V.WCHK,J/S05420 ; (2)
```

```
U 102F, 0000,003D,0180,0800,0000,113D  ;3616  ERROR1    ;SHOULD NEVER REACH HERE
        ;3617 =0
        ;3618 S05420: D_K[.FF],
U 10E2, 0818,0039,4980,0800,0000,11EA  ;3619   CALL[WAIT.LOOP]  ;LET SILO FILL AND WAIT
        ;3620       ; ...FOR THE INTERLOCK FLIP-FLOP TO
        ;3621       ; ...CLEAR
U 10E3, 0000,003C,0180,0800,0000,1030  ;3622  NOP
U 1030, 0000,003D,0180,0800,0000,11D1  ;3623 =00 CALL[SILINT]
U 1031, 0000,003D,0180,0800,0000,113D  ;3624  ERROR1    ;DIDN'T GET SILO
        ;3625       ; INTERRUPT
        ;3626   ;=============================================================MS
        ;3627   ;
        ;3628   ;CHECK SILO CONTENTS
        ;3629   ;=============================================================
U 1032, 0F03,003C,0180,09D0,0000,1033  ;3630  RC[0A]_0,D_0   ;RESET LOOP COUNTER
U 1033, 0003,003C,0180,09E0,0000,128E  ;3631  RC[0C]_0   ;SET EXPECTED DATA
        ;3632   ;=============================================================
        ;3633   ;SEE NULL CYCLES
        ;3634   ;=============================================================
U 128E, 0000,003C,61F0,2C00,0000,128F  ;3635 S05430: Q_ID[SILO]
U 128F, 0001,203C,0180,09D8,0010,1290  ;3636  RC[0B]_Q,CLK.UBCC
U 1290, 0810,0138,0180,0950,0000,10E4  ;3637  Z?,D_RC[0A]
U 10E4, 0000,003D,0180,0800,0000,113D  ;3638 =0 ERROR1    ;DIDN'T SEE NULL CYCLE
U 10E5, 0019,0020,0D80,0800,0010,1291  ;3639  ALU_D[XOR]K[.3],CLK.UBCC
U 1291, 0819,0114,0580,09D0,0000,10E6  ;3640  Z?,RC[0A]_ALU,D_D+K[.1]
U 10E6, 0000,003C,0180,0800,0000,128E  ;3641 =0 J/S05430   ;REPEAT LOOP
        ;3642   ;=============================================================
        ;3643   ;SEE CPU ARB IF I.D. NOT = 10
        ;3644   ;=============================================================
U 10E7, 0010,0038,01C0,0910,0000,1292  ;3645  Q_RC[2]
U 1292, 0019,2020,6580,0800,0010,1293  ;3646  ALU_Q.XOR.K[.10],CLK.UBCC
U 1293, 0000,013C,0180,0800,0000,10E8  ;3647  Z?
U 10E8, 0000,003C,0180,0800,0000,1294  ;3648 =0 J/S05440   ;=D NOT 10
U 10E9, 0000,003C,0180,0800,0000,1299  ;3649  J/S05450   ;ID =10,SKIP ARB
U 1294, 0800,003C,61F0,2E50,0000,1295  ;3650 S05440: D_R[0A],Q_ID[SILO]
U 1295, 0001,003C,0180,09E0,0000,1296  ;3651  RC[0C]_D
U 1296, 0001,203C,0180,09D8,0000,1297  ;3652  RC[0B]_Q
U 1297, 001D,0020,0180,0800,0010,1298  ;3653  ALU_D[XOR]Q,CLK.UBCC
U 1298, 0810,0138,0180,0950,0000,10EA  ;3654  Z?,D_RC[0A]
U 10EA, 0000,003D,0180,0800,0000,113D  ;3655 =0 ERROR1    ;BAD CPU ARB IN SILO
        ;3656   ;=============================================================
        ;3657   ;SEE C/A (INTERLOCK READ)
        ;3658   ;=============================================================
U 10EB, 0819,0014,0580,09D0,0000,1299  ;3659  RC[0A]_ALU,D_D+K[.1]
U 1299, 0800,003C,61F0,2E08,0000,129A  ;3660 S05450: D_R[1],Q_ID[SILO]
U 129A, 0001,003C,0180,09E0,0000,129B  ;3661  RC[0C]_D
U 129B, 0001,203C,0180,09D8,0000,129C  ;3662  RC[0B]_Q
U 129C, 001D,0020,0180,0800,0010,129D  ;3663  ALU_D[XOR]Q,CLK.UBCC
U 129D, 0810,0138,0180,0950,0000,10EC  ;3664  Z?,D_RC[0A]
U 10EC, 0000,003D,0180,0800,0000,113D  ;3665 =0 ERROR1    ;BAD C/A IN SILO
        ;3666   ;=============================================================
        ;3667   ;SEE NULL (INTERLOCK BIT DRIVEN BY CPU)
        ;3668   ;=============================================================
U 10ED, 0819,0014,0580,09D0,0000,129E  ;3669  RC[0A]_ALU,D_D+K[.1]
U 129E, 0800,003C,0180,0A48,0000,129F  ;3670  D_R[9]    ;SEE INTERLOCK DRIVEN
```

```
U 129F. 0001.003C.61F0.2DE0.0000.12A0  ;3671  RC[0C]_D,Q_ID[SILO]
U 12A0. 0001.203C.0180.09D8.0000.12A1  ;3672  RC[0B]_Q
U 12A1. 001D.0020.0180.0800.0010.12A2  ;3673  ALU_D[XOR]Q,CLK.UBCC
U 12A2. 0810.0138.0180.0950.0000.10EE  ;3674  Z?,D_RC[0A]
U 10EE. 0000.003D.0180.0800.0000.113D  ;3675 =0 ERROR1     ;NULL DATA BAD IN SILO
          ;3676   ;================================================================
          ;3677   ;SEE ACK FROM MEMORY
          ;3678   ;================================================================
U 10EF. 0819.0014.0580.09D0.0000.12A3  ;3679  RC[0A]_ALU,D_D+K[.1]
U 12A3. 0010.0038.0180.0940.0010.12A4  ;3680  ALU_rc[8],CLK.UBCC
U 12A4. 0F00.013C.0180.0800.0000.10F0  ;3681  Z?,D_0 ; Mod for D_R[9]   ;NON I/O REFERENCE
U 10F0. 0F00.003C.0180.0800.0000.10F1  ;3682 =0 D_0     ;I/O REFERENCE
U 10F1. 080D.0030.0180.0A68.0000.12A5  ;3683  D_ALU,ALU_D[OR]R[0D]
U 12A5. 0001.003C.61F0.2DE0.0000.12A6  ;3684  RC[0C]_D,Q_ID[SILO]
U 12A6. 0001.203C.0180.09D8.0000.12A7  ;3685  RC[0B]_Q
U 12A7. 001D.0034.01C0.0800.0000.12A8  ;3686  Q_D[AND]Q   ; Mask unwanted bits
U 12A8. 001D.0020.0180.0800.0010.12A9  ;3687  ALU_D[XOR]Q,CLK.UBCC
U 12A9. 0000.013C.0180.0800.0000.10F2  ;3688  Z?
U 10F2. 0000.003D.0180.0800.0000.113D  ;3689 =0 ERROR1     ;ACK IN SILO BAD
          ;3690   ;================================================================
          ;3691   ;SEE MEMORY ARB ON BUS
          ;3692   ;================================================================
U 10F3. 0010.0038.0180.0940.0010.12AB  ;3693  ALU_rc[8],CLK.UBCC
U 12AB. 0F00.013C.0180.0800.0000.10F4  ;3694  Z?,D_0 ; Mod for D_R[9]   ;NON I/O REFERENCE
U 10F4. 0F00.003C.0180.0800.0000.10F5  ;3695 =0 D_0     ;I/O REFERENCE
U 10F5. 0000.003C.91F0.2C00.0000.12AC  ;3696  Q_ID[24]   ; GET MEMORY ID LEVEL
U 12AC. 0001.203C.0180.0800.0082.12AD  ;3697  SC_Q
U 12AD. 0801.001C.0180.0800.0000.12AE  ;3698  D_D.ORNOT.MASK   ; SET MEMORY TR LEVEL
U 12AE. 0001.003C.0180.09E0.0000.12AF  ;3699  RC[0C]_D
U 12AF. 0010.0038.01C0.0958.0000.12B0  ;3700  Q_RC[0B]   ; Get SILO contents
U 12B0. 001D.0034.01C0.0800.0000.12B1  ;3701  Q_D[AND]Q   ; Mask unwanted bits
U 12B1. 001D.0020.0180.0800.0010.12B2  ;3702  ALU_D[XOR]Q,CLK.UBCC
U 12B2. 0810.0138.0180.0950.0000.10F6  ;3703  Z?,D_RC[0A]
U 10F6. 0000.003D.0180.0800.0000.113D  ;3704 =0 ERROR1     ;MEMORY ARB IN SILO BAD
          ;3705   ;================================================================
          ;3706   ;SEE READ DATA IN SILO
          ;3707   ;================================================================
U 10F7. 0819.0014.0580.09D0.0000.12B3  ;3708  RC[0A]_ALU,D_D+K[.1]
U 12B3. 0010.0038.0180.0940.0010.12B4  ;3709  ALU_rc[8],CLK.UBCC
U 12B4. 0F00.013C.0180.0800.0000.10F8  ;3710  Z?,D_0 ; Mod for D_R[9]   ;NON I/O REFERENCE
U 10F8. 0F00.003C.0180.0800.0000.10F9  ;3711 =0 D_0     ;I/O REFERENCE
U 10F9. 080D.0030.0180.0A30.0000.12B5  ;3712  D_ALU,ALU_D[OR]R[6]
U 12B5. 0001.003C.61F0.2DE0.0000.12B6  ;3713  RC[0C]_D,Q_ID[SILO]
U 12B6. 0001.203C.0180.09D8.0000.12B7  ;3714  RC[0B]_Q
U 12B7. 001D.0020.0180.0800.0010.12B8  ;3715  ALU_D[XOR]Q,CLK.UBCC
U 12B8. 0810.0138.0180.0950.0000.10FA  ;3716  Z?,D_RC[0A]
U 10FA. 0000.003D.0180.0800.0000.113D  ;3717 =0 ERROR1     ;READ DATA IN SILO BAD
          ;3718   ;================================================================
          ;3719   ;SEE NULL CYCLE
          ;3720   ;================================================================
U 10FB. 0819.0014.0580.09D0.0000.12B9  ;3721  RC[0A]_ALU,D_D+K[.1]
U 12B9. 0010.0038.0180.0940.0010.12BA  ;3722  ALU_rc[8],CLK.UBCC
U 12BA. 0F00.013C.0180.0800.0000.10FC  ;3723  Z?,D_0 ; Mod for D_R[9]   ;NON I/O REFERENCE
U 10FC. 0F00.003C.0180.0800.0000.10FD  ;3724 =0 D_0     ;I/O REFERENCE
U 10FD. 0001.003C.61F0.2DE0.0000.12BB  ;3725  RC[0C]_D,Q_ID[SILO]
```

```
U 12BB. 0001.203C.0180.09D8.0000.12BC  ;3726  RC[0B]_Q
U 12BC. 001D.0020.0180.0800.0010.12BD  ;3727  ALU_D[XOR]Q,CLK.UBCC
U 12BD. 0810.0138.0180.0950.0000.10FE  ;3728  Z?,D_RC[0A]
U 10FE. 0000.003D.0180.0800.0000.113D  ;3729 =0 ERROR1     ;NULL CYCLE BAD IN SILO
          ;3730    ;=============================================================
          ;3731    ;SEE CPU ACK TO MEMORY
          ;3732    ;=============================================================
U 10FF. 0819.0014.0580.09D0.0000.12BE  ;3733  RC[0A]_ALU,D_D+K[.1]
U 12BE. 0010.0038.0180.0940.0010.12BF  ;3734  ALU_rc[8],CLK.UBCC
U 12BF. 0F00.013C.0180.0800.0000.110A  ;3735  Z?,D_0 ; Mod for D_R[9]    ;NON I/O REFERENCE
U 110A. 0F00.003C.0180.0800.0000.110B  ;3736 =0 D_0     ;I/O REFERENCE
U 110B. 080D.0030.0180.0A68.0000.12C0  ;3737  D_ALU,ALU_D[OR]R[0D]
U 12C0. 0001.003C.61F0.2DE0.0000.12C1  ;3738  RC[0C]_D,Q_ID[SILO]
U 12C1. 0001.203C.0180.09D8.0000.12C2  ;3739  RC[0B]_Q
U 12C2. 001D.0020.0180.0800.0010.12C3  ;3740  ALU_D[XOR]Q,CLK.UBCC
U 12C3. 0810.0138.0180.0950.0000.1110  ;3741  Z?,D_RC[0A]
U 1110. 0000.003D.0180.0800.0000.113D  ;3742 =0 ERROR1     ;CPU ACK BAD FROM SILO
          ;3743    ;=============================================================
          ;3744    ;SEE NULL UNTIL LOC COUNTER = F
          ;3745    ;=============================================================
U 1111. 0819.0014.0580.09D0.0000.12C4  ;3746 S05460: RC[0A]_ALU,D_D+K[.1]
U 12C4. 0010.0038.0180.0940.0010.12C5  ;3747  ALU_rc[8],CLK.UBCC
U 12C5. 0F00.013C.0180.0800.0000.1112  ;3748  Z?,D_0 ; Mod for D_R[9]    ;NON I/O REFERENCE
U 1112. 0F00.003C.0180.0800.0000.1113  ;3749 =0 D_0     ;I/O REFERENCE
U 1113. 0001.003C.61F0.2DE0.0000.12C6  ;3750  RC[0C]_D,Q_ID[SILO]
U 12C6. 0001.203C.0180.09D8.0000.12C7  ;3751  RC[0B]_Q
U 12C7. 001D.0020.0180.0800.0010.12C8  ;3752  ALU_D[XOR]Q,CLK.UBCC
U 12C8. 0810.0138.0180.0950.0000.1114  ;3753  Z?,D_RC[0A]
U 1114. 0000.003D.0180.0800.0000.113D  ;3754 =0 ERROR1     ;NULL CYCLE BAD
U 1115. 0019.0020.6180.0800.0000.12C9  ;3755  ALU_D[XOR]K[.F]
U 12C9. 0000.013C.0180.0800.0000.1116  ;3756  Z?
U 1116. 0000.003C.0180.0800.0000.1111  ;3757 =0 J/S05460    ;REPEAT LOOP
          ;3758    ;=============================================================
          ;3759    ;CHECK OPERATION POINTER AND ADDRESS FLAG
          ;3760    ;=============================================================
U 1117. 0810.0038.0180.0948.0000.12CA  ;3761  D_RC[9]    ;CHECK OPERATION
U 12CA. 0019.0020.0980.0800.0010.12CB  ;3762  ALU_D.XOR.K[.2],CLK.UBCC ; POINTER
U 12CB. 0000.013C.0180.0800.0000.1118  ;3763  Z?
U 1118. 0819.0014.0580.09C8.0000.10D9  ;3764 =0 RC[9]_ALU,D_D+K[.1],J/S05410
U 1119. 0010.0038.0180.0940.0010.12CC  ;3765  ALU_rc[8],CLK.UBCC  ;CHECK ADDRESS FLAG
U 12CC. 0000.013C.0180.0800.0000.111A  ;3766  Z?
U 111A. 0000.003C.0180.0800.0000.12CD  ;3767 =0 J/S05END
U 111B. 0018.0038.0580.09C0.0000.128B  ;3768  rc[8]_K[.1],J/S05400
U 12CD. 0000.003C.0180.0800.0000.111C  ;3769 S05END: NOP    ; ALL DONE.
          ;3770
          ;3771
```

```
            ;3772  .PAGE "TEST 19E RESERVED"
            ;3773 =0
U 111C, 0000,003D,0180,0800,0000,1127  ;3774 T1A1: NEWTST
U 111D, 0000,003C,0180,0800,0000,111E  ;3775  NOP
            ;3776
```

```
            ;3777  .PAGE "TEST 19F RESERVED"
            ;3778 =0
U 111E. 0000,003D,0180,0800,0000,1127  ;3779 T19F: NEWTST
U 111F. 0000,003C,0180,0800,0000,1120  ;3780  NOP
            ;3781
```

```
            ;3782  .PAGE "TEST 1A0 RESERVED"
            ;3783 =0
U 1120, 0000,003D,0180,0800,0000,1127  ;3784 T1A0: NEWTST
U 1121, 0000,003C,0180,0800,0000,12CE  ;3785  NOP
            ;3786
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 0581
ESKAR43.MCR                    MICRO2  1P(00)    29-JUL-85  15:01:16          Page  11

            ;3787  .PAGE "DUMMY NEWTST"
U 12CE. 0000,003D,0180,0800,0000,1127  ;3788 DUM: NEWTST
            ;3789
```

```
;                       Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1912: 1879 1886= 1902= 2634= 2635= 2636= 2638=
U 1008  2357= 2358= 2359= 2360= 2361= 2362= 2363= 2364=
U 1010  2682= 2683= 2684= 2686= 3152= 3153= 3154= 1880
U 1018  3167= 3169= 3170= 1881 1903= 1904= 3407= 3408=
U 1020  3175= 3180= 3194= 1882 3259= 3260= 3261= 3262=
U 1028  3413= 3414= 3420= 1883 3613= 3614= 3615= 3616=
U 1030  3623= 3624= 3630= 3631= 1956= 1958= 1969= 1970=
U 1038  1999= 2000= 2026= 2027= 2132= 2133= 2136= 2137=
U 1040  2139= 2140= 2144= 2145= 2217= 2218= 2292= 2293=
U 1048  2336= 2337= 2345= 2346= 2347= 2349= 1980: 1884
U 1050  2352= 2353= 2394= 2395= 2396= 2397= 2404= 2405=
U 1058  2411= 2413= 2008: 1885 2416= 2417= 2035: 1905
U 1060  2436= 2437= 2439= 2440= 2532= 2533= 2536= 2537=
U 1068  2548= 2549= 2552= 2553= 2589= 2590= 2703= 2704=
U 1070  2706= 2707= 2709= 2710= 2730= 2731= 2733= 2734=
U 1078  2740= 2741= 2764= 2765= 2766= 2768= 2769= 2770=
U 1080  2771= 2772= 2773= 2774= 2775= 2776= 2777= 2778=
U 1088  2780= 2781= 2817= 2818= 2829= 2830= 2929= 2930=
U 1090  3007= 3008= 3010= 3011= 3149= 3151= 3220= 3221=
U 1098  3231= 3232= 3233= 3234= 3243= 3244= 3256= 3257=
U 10A0  3272= 3273= 3292= 3293= 3387= 3388= 3390= 3392=
U 10A8  3393= 3394= 3396= 3397= 3401= 3404= 3411= 3412=
U 10B0  3422= 3426= 3440= 3441= 3445= 3446= 3448= 3449=
U 10B8  3452= 3453= 3459= 3460= 3465= 3470= 3473= 3474=
U 10C0  3480= 3484= 3487= 3488= 3493= 3497= 3501= 3502=
U 10C8  3504= 3505= 3510= 3511= 3514= 3518= 3522= 3523=
U 10D0  3525= 3531= 3534= 3535= 3537= 3538= 3597= 3598=
U 10D8  3600= 3602= 3603= 3604= 3605= 3606= 3608= 3609=
U 10E0  3610= 3611= 3619= 3622= 3638= 3639= 3641= 3645=
U 10E8  3648= 3649= 3655= 3659= 3665= 3669= 3675= 3679=
U 10F0  3682= 3683= 3689= 3693= 3695= 3696= 3704= 3708=
U 10F8  3711= 3712= 3717= 3721= 3724= 3725= 3729= 3733=
U 1100  1866: 1867: 1868: 1869: 1870: 1871: 1872: 1873:
U 1108  1874: 1906 3736= 3737= 1875: 1876: 1877: 1878:
U 1110  3742= 3746= 3749= 3750= 3754= 3755= 3757= 3761=
U 1118  3764= 3765= 3767= 3768= 3774= 3775= 3779= 3780=
U 1120  3784= 3785= 1907 1908 1909 1910 1911 1945
U 1128  1946 1947 1948 1949 1950 1951 1952 1953
U 1130  1954 1955 1959 1960 1961 1962 1963 1964
U 1138  1965 1966 1967 1968 1974 1978 1979 2006
U 1140  2007 2069 2070 2080 2081 2090 2091 2092
U 1148  2105 2106 2107 2108 2109 2134 2135 2141
U 1150  2142 2143 2146 2147 2148 2058: 2149 2150
U 1158  2151 2152 2153 2173 2175 2176 2177 2178
U 1160  2179 2184 2210 2211 2212 2213 2215 2216
U 1168  2245 2246 2247 2266 2267 2268 2269 2294
U 1170  2295 2296 2297 2338 2339 2340 2341 2350
U 1178  2351 2355 2356 2369 2370 2372 2373 2374
U 1180  2376 2381 2382 2383 2385 2390 2392 2393
U 1188  2401 2402 2406 2407 2408 2409 2410 2418
U 1190  2419 2420 2422 2423 2424 2425 2426 2427
U 1198  2432 2434 2435 2438 2463 2464 2465 2466
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2496 2497 2499 2500 2501 2528 2530 2531
U 11A8  2535 2539 2540 2541 2542 2544 2545 2546
U 11B0  2547 2550 2551 2578 2579 2580 2581 2582
U 11B8  2583 2584 2585 2587 2588 2701 2732 2735
U 11C0  2736 2737 2738 2739 2742 2749 2750 2751
U 11C8  2763 2782 2783 2811 2812 2813 2814 2815
U 11D0  2816 2823 2824 2825 2826 2827 2828 2842
U 11D8  2843 2844 2855 2856 2857 2858 2869 2870
U 11E0  2871 2872 2883 2884 2885 2886 2896 2897
U 11E8  2898 2899 2924 2952 2994 2996 2997 2998
U 11F0  2999 3000 3003 3005 3006 3157 3158 3159
U 11F8  3161 3163 3164 3165 3166 3174 3222 3223
U 1200  3224 3225 3226 3230 3235 3236 3237 3238
U 1208  3239 3240 3241 3242 3246 3247 3248 3249
U 1210  3250 3251 3252 3253 3258 3263 3264 3265
U 1218  3266 3267 3268 3269 3270 3271 3274 3275
U 1220  3276 3277 3278 3279 3280 3281 3282 3283
U 1228  3284 3285 3286 3287 3288 3289 3290 3291
U 1230  3294 3295 3296 3297 3298 3299 3300 3301
U 1238  3302 3303 3304 3305 3306 3307 3308 3309
U 1240  3310 3311 3312 3313 3314 3315 3316 3317
U 1248  3318 3319 3320 3321 3322 3323 3324 3325
U 1250  3326 3327 3328 3329 3330 3331 3332 3333
U 1258  3334 3335 3389 3395 3405 3406 3421 3439
U 1260  3442 3443 3444 3447 3450 3451 3457 3458
U 1268  3461 3462 3463 3464 3471 3472 3475 3476
U 1270  3477 3478 3479 3485 3486 3489 3490 3491
U 1278  3492 3498 3499 3500 3503 3506 3507 3508
U 1280  3509 3512 3513 3519 3520 3521 3524 3532
U 1288  3533 3536 3540 3599 3607 3612 3635 3636
U 1290  3637 3640 3646 3647 3650 3651 3652 3653
U 1298  3654 3660 3661 3662 3663 3664 3670 3671
U 12A0  3672 3673 3674 3680 3681 3684 3685 3686
U 12A8  3687 3688 2059: 3694 3697 3698 3699 3700
U 12B0  3701 3702 3703 3709 3710 3713 3714 3715
U 12B8  3716 3722 3723 3726 3727 3728 3734 3735
U 12C0  3738 3739 3740 3741 3747 3748 3751 3752
U 12C8  3753 3756 3762 3763 3766 3769 3788
U 12D0 - 13EF Unused
U 13F0  2042: 2043: 2044: 2045: 2046: 2047: 2048: 2049:
U 13F8  2050: 2051: 2052: 2053: 2054: 2055: 2056: 2057:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0584
ESKAR43.MCR                    MICRO2  1P(00)    29-JUL-85  15:01:16     Page  12

          Words not
          in bounds
 ESK00DEF         0
 ESK00MAC         0
 ESKARMAC         0
  ESKAR43       735

Used            735
Remaining

Total microwords used in memory U: 735
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0585
ESKAR43.MCR                    MICRO2  1P(00)    29-JUL-85  15:01:16     Page  12


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR43.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 0586
ESKAR44.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1917 Micro Monitor Interface Routines
; 1918   Newtest Routine
; 1975   Disable Cache Routine
; 1997   ERLOOP ROUTINE
; 2000   ERROR 1 ROUTINE
; 2005   ERROR1 WITH PARAMETER
; 2026   ERROR 2 ROUTINE
; 2031   ERROR 2 WITH PARAMETER
; 2051   CONSOLE CALL ROUTINE
; 2056   SCRATCH WCS LOCATIONS
; 2079   RESET SUBTEST NUMBER
; 2101   SUBTEST SUBROUTINE
; 2111   SUBTEST 1 ROUTINE
; 2121   SETRCF ROUTINE
; 2132   FIND MS780-E MEMORY
; 2274   Generate IO Address
; 2300   Set Starting Address
; 2335   ENABLE NEXT MS780-E
; 2387   Set Trap Mask
; 2417   CHECK UTRAP
; 2448   Initialize the SBI
; 2477   Select Controller
; 2515   SELECT LOWER CONTROLLER
; 2564   SELECT UPPER CONTROLLER
; 2613   CONFIGURE MS780-E/H
; 2687   ENBFLT ROUTINE
; 2706   NOINTR ROUTINE
; 2789   DELAY ROUTINE
; 2795   Wait Loop Routine
; 2827   SBI UNJAM (SBI CLEAR) ROUTINE
; 2849   INIT ROUTINE
; 2979 TEST 1A1 MISCELLANEOUS SILO AND SBI (CPU) INTERFACE TESTS
; 3963 TEST 1A2 SBI DISABLE
; 4099 TEST 1A3 RESERVED
; 4104 TEST 1A4 RESERVED
; 4109 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR44
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.05B0.0800.0084.7001  ;1862 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.09B0.0800.0084.7001  ;1863 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0DB0.0800.0084.7001  ;1864 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.11B0.0800.0084.7001  ;1865 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D5B0.0800.0084.7001  ;1866 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D9B0.0800.0084.7001  ;1867 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5DB0.0800.0084.7001  ;1868 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7001  ;1869 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7001  ;1870 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D. 0000.003C.8980.0800.0084.7001  ;1871 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7001  ;1872 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1017  ;1874 TRPH0: Q_ID[USTACK]
U 1017. 0C00.003C.01E0.0800.0000.101B  ;1875  Q_D,D_Q
U 101B. 0000.003C.8180.3C00.0000.101F  ;1876  ID[USTACK]_D
U 101F. 0C00.003C.01E0.0800.0000.102B  ;1877  Q_D,D_Q
U 102B. 0000.003C.01C0.0A78.0000.102F  ;1878  Q_R[0F]
U 102F. 0001.2024.0180.0800.0010.104F  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 104F. 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.104C  ;1897 TRPH1: RC[0D]_SC
U 104C. 0000.003D.D980.0800.0084.71C4  ;1898 =0 SETRCF[.9]
U 104D. 0000.003C.49F0.2C00.0000.1057  ;1899  Q_ID[TBER0]
U 1057. 0001.203C.4DF0.2DB0.0000.105B  ;1900  RC[6]_Q,Q_ID[TBER1]
U 105B. 0001.203C.65F0.2DB8.0000.105F  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 105F. 0001.203C.6DF0.2DD8.0000.1063  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1063. 0001.203C.75F0.2DE0.0000.1109  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1109. 0001.203C.79F0.2DC8.0000.1154  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1154. 0001.203C.69F0.2DC0.0000.11A0  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 11A0. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.11BB  ;1907 1000: RC[0E]_Q,ERROR1
```

```
                ;1908  .PAGE
                ;1909 ;*
                ;1910 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1911 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1912 ;  NEWTST
                ;1913 ;  ERLOOP
                ;1914 ;  ERROR1
                ;1915 ;  ERROR2
                ;1916 ;
                ;1917  .TOC " Micro Monitor Interface Routines"
                ;1918  .TOC "   Newtest Routine"
                ;1919 ;**
                ;1920 ; FUNCTIONAL DESCRIPTION:
                ;1921 ;
                ;1922 ; This routine Initializes the following registers:
                ;1923 ;  PSL   to 1F
                ;1924 ;  CES   to 0
                ;1925 ;  ACC.CS   to disable accellerator
                ;1926 ;  SIR   to 0
                ;1927 ;  CLK.CS   to stop interval timer
                ;1928 ;  TBER0   to disable the TB
                ;1929 ;  SBI.MAINT to    0
                ;1930 ;  SBI.ERR   to 0
                ;1931 ;  FAULT   to 0
                ;1932 ;  COMP   to 0
                ;1933 ;  RC[0E]   to  0
                ;1934 ;  R[0F]   to 0
                ;1935 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1936 ; A SCOPE call is then made to the Monitor.
                ;1937 ;
                ;1938 ; CALLING CONSTRAINT:
                ;1939 ;
                ;1940 ; =0
                ;1941 ;
                ;1942 ; SIDE EFFECTS:
                ;1943 ;
                ;1944 ; D Reg is destroyed
                ;1945 ; SC is destroyed
                ;1946 ;--
U 11A1, 0000,003C,65F8,0800,0084,71A2  ;1947 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 11A2, 0818,0038,8D80,0800,0000,11A3  ;1948  d_k[.1f]  ; D=1F
U 11A3, 0D00,003C,0180,0800,0000,11A4  ;1949  d_dal.sc  ; D=001F0000
U 11A4, 0000,003C,3D80,3C00,0000,11A5  ;1950  id[psl]_d  ; psl = 001F0000
U 11A5, 0818,0038,0580,0800,0000,11A6  ;1951  d_k[.1]  ; Clear the CES register
U 11A6, 0D00,003C,0180,0800,0000,11A7  ;1952  d_dal.sc  ; D = 00010000
U 11A7, 0F00,003C,3180,3C00,0000,11A8  ;1953  id[ces]_d,d_0  ; ces = 00010000
U 11A8, 0000,003C,5D80,3C00,0000,11A9  ;1954  id[acc.cs]_d  ; acc.cs = 0
U 11A9, 0000,003C,3980,3C00,0000,11AA  ;1955  id[sir]_d  ; sir = 0
U 11AA, 0000,003C,2980,3C00,0000,11AB  ;1956  id[clk.cs]_d  ; clk.cs = 0
U 11AB, 0000,003C,4980,3C00,0000,1058  ;1957  id[tber0]_d  ; tber0 = 0
U 1058, 0000,003D,0180,0800,0000,124C  ;1958 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1959  intrpt.strobe,  ; Strobe interrupts
U 1059, 0818,0038,C180,0800,4000,11AC  ;1960      d_k[.ffff]  ; Setup to clear SBI error register and
U 11AC, 0801,0028,6580,3C00,0000,11AD  ;1961  id[sbi.err]_d,d_not.d ; and fault register
U 11AD, 0F00,003C,6D80,3C00,0000,11AE  ;1962  id[fault]_d,d_0  ; ...
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0593
ESKAR44.MCR                        MICRO2  1P(00)    26-JUL-85  17:12:03                 Page   5

U 11AE, 0000,003C,6D80,3C00,0000,11AF  ;1963  id[fault]_d  ; fault = 0
U 11AF, 0000,003C,7580,3C00,0000,11B0  ;1964  id[maint]_d  ; MAINT = 0
U 11B0, 0000,003C,6580,3C00,0000,11B1  ;1965  id[sbi.err]_d  ; sbi error reg = 0
U 11B1, 0000,003C,7180,3C00,0000,11B2  ;1966  id[comp]_d  ; comp reg = 0
U 11B2, 0000,003C,E580,3C00,0000,11B3  ;1967  id[T9]_d  ; Clear code for subroutine START.TEST
U 11B3, 0001,003C,0180,09F0,0000,11B4  ;1968  rc[0e]_d  ;
U 11B4, 0001,003C,0180,0AF8,0000,11B5  ;1969  r[0f]_d  ; Clear expected trap mask
U 11B5, 0001,003C,0180,09F8,0000,105C  ;1970  rc[0f]_d  ; Clear subtest number and argumnet count
U 105C, 0000,003D,0180,0800,0000,11B6  ;1971 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1972  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1973
```

```
            ;1974  .PAGE
            ;1975  .TOC "   Disable Cache Routine"
            ;1976 ;++
            ;1977 ; FUNCTIONAL DESCRIPTION:
            ;1978 ;
            ;1979 ; This routine disables cache hits by setting bits <16:15>
            ;1980 ; in the maintenance register.
            ;1981 ;
            ;1982 ; CALLING SEQUENCE:
            ;1983 ;
            ;1984 ; =0
            ;1985 ;
            ;1986 ; SIDE EFFECTS:
            ;1987 ;
            ;1988 ; D & Q Registers destroyed
            ;1989 ;--
            ;1990 DISABLE.CACHE:
U 11B6, 0818.0038,4580,0800,0000,11B7  ;1991  d_k[.8000]  ; ... and seed constant
U 11B7, 0500.003C,0180,0800,0000,11B8  ;1992  d_d.left  ; Generate final constant
U 11B8, 0819.0030,4580,0800,0000,11B9  ;1993  d_d.or.k[.8000]  ; ... = 00018000
U 11B9, 0000.003E,7580,3C00,0000,0001  ;1994  id[maint]_d,return1 ; Disable cache and return
            ;1995
```

```
            ;1996  .PAGE
            ;1997  .TOC "   ERLOOP ROUTINE"
U 11BA, 0818,0038,0980,0800,0000,105E  ;1998 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
```

```
          ;1999  .PAGE
          ;2000  .TOC "   ERROR 1 ROUTINE"
U 11BB, 0810,0038,0180,0978,0000,11BC  ;2001 ERROR1: D_RC[0F]
U 11BC, 0819,0034,4D80,0800,0000,104E  ;2002  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2003 104E: D_D.OR.K[.4],J/CALLCON
```

```
            ;2004  .PAGE
            ;2005  .TOC "   ERROR1 WITH PARAMETER"
            ;2006 ;**
            ;2007 ; FUNCTIONAL DESCRIPTION:
            ;2008 ;
            ;2009 ; This subroutine takes  as parameter the number of arguments
            ;2010 ; to be printed to the console in the case of an error logout.
            ;2011 ;
            ;2012 ; CALLING CONSTRAINT:
            ;2013 ;
            ;2014 ; =0
            ;2015 ;INPUTS:
            ;2016 ;
            ;2017 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2018 ;--
            ;2019 =0
            ;2020 ERR1.ARG:
U 1064, 0000,003D,0180,0800,0000,11C4  ;2021  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1065, 0000,003C,0180,0800,0000,11BB  ;2022  J/ERROR1  ; Go to ERROR1
            ;2023 ;
            ;2024 ;
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0598
ESKAR44.MCR                    MICRO2  1P(00)    26-JUL-85  17:12:03                    Page   5

            ;2025  .PAGE
            ;2026  .TOC "    ERROR 2 ROUTINE"
U 11BD, 0810,0038,0180,0978,0000,11BE  ;2027 ERROR2: D_RC[0F]
U 11BE, 0819,0034,4D80,0800,0000,105A  ;2028  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2029 105A: D_D.OR.K[.6],J/CALLCON
```

```
            ;2030  .PAGE
            ;2031  .TOC "    ERROR 2 WITH PARAMETER"
            ;2032 ;++
            ;2033 ; FUNCTIONAL DESCRIPTION:
            ;2034 ;
            ;2035 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2036 ; except that in this case after setting the number of arguments
            ;2037 ; too the console, control is transferred to routine ERROR2.
            ;2038 ;
            ;2039 ; INPUTS:
            ;2040 ;
            ;2041 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2042 ;
            ;2043 ;--
            ;2044 =0
            ;2045 ERR2.ARG:
U 1066. 0000,003D,0180,0800,0000,11C4  ;2046  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1067. 0000,003C,0180,0800,0000,11BD  ;2047  J/ERROR2  ; Go to ERROR2
            ;2048 ;
            ;2049 ;
```

```
            ;2050  .PAGE
            ;2051  .TOC "    CONSOLE CALL ROUTINE"
            ;2052 105E:
            ;2053 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2054  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
```

```
            ;2055  .PAGE
            ;2056  .TOC "   SCRATCH WCS LOCATIONS"
            ;2057 ;*
            ;2058 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2059 ;-
U 13F0, 0000,003C,0180,0800,0000,13F1   ;2060 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2   ;2061 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3   ;2062 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4   ;2063 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5   ;2064 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6   ;2065 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7   ;2066 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8   ;2067 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9   ;2068 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA   ;2069 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB   ;2070 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC   ;2071 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD   ;2072 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE   ;2073 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF   ;2074 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155   ;2075 13FF: NOP
U 1155, 0000,003C,0180,0800,0000,12AA   ;2076 1155: NOP
U 12AA, 0000,003C,0180,0800,0000,11BF   ;2077 12AA: NOP
```

```
            ;2078  .PAGE
            ;2079  .TOC "   RESET SUBTEST NUMBER"
            ;2080 ;**
            ;2081 ; FUNCTIONAL DESCRIPTION:
            ;2082 ;
            ;2083 ; Since some  of the tests will have to be restarted when two
            ;2084 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2085 ; the subtest counter to zero ( only for thoes tests that have
            ;2086 ; more than one subtest). This subroutine may also be necessary
            ;2087 ; when a test is being loop on.
            ;2088 ;
            ;2089 ; CALLING CONSTRAINT:
            ;2090 ;
            ;2091 ; =0
            ;2092 ; SIDE EFFECTS:
            ;2093 ;
            ;2094 ; D is destroyed
            ;2095 ;
            ;2096 ;--
            ;2097 RESET.SUBTST:
            ;2098  RC[0F] 0,   ; Reset subtest number
U 11BF, 0003,003E,0180,09F8,0000,0001  ;2099   RETURN1   ; ...and return
```

```
            ;2100   .PAGE
            ;2101   .TOC "    SUBTEST SUBROUTINE"
            ;2102 ;*
            ;2103 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2104 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2105 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2106 ; TYPING IT.
            ;2107 ;-
U 11C0, 0810.0038.4D80.0978.0000.11C1   ;2108 SUBTST: D_RC[0F],KMX/.FF00
U 11C1, 0019.4016.4D80.09F8.0000.0001   ;2109   RC[0F]_D+K[.FF00],WORD,RETURN1
```

```
            ;2110   .PAGE
            ;2111   .TOC "    SUBTEST 1 ROUTINE"
            ;2112 ;*****************************************************************
            ;2113 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2114 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2115 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2116 ;-
            ;2117 SUBTEST1:
U 11C2, 0810.0038.0180.0978.0000.11C3  ;2118  D_RC[0F]
U 11C3, 0019.0032.4D80.09F8.0000.0001  ;2119  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
```

```
            ;2120  .PAGE
            ;2121  .TOC "   SETRCF ROUTINE"
            ;2122 ;*
            ;2123 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2124 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2125 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2126 ;-
U 11C4, 0010,0038,01C0,0978,0000,11C5  ;2127 SETRCF: Q_RC[0F]
U 11C5, 0019,2034,4DC0,0800,0000,11C6  ;2128  Q_Q.AND.K[.FF00]
U 11C6, 0019,2032,1D80,09F8,0000,0001  ;2129  RC[0F]_Q.OR.SC,RETURN1
            ;2130
```

```
            ;2131  .PAGE
            ;2132  .TOC "    FIND MS780-E MEMORY"
            ;2133 ;**
            ;2134 ; FUNCTIONAL DESCRIPTION:
            ;2135 ;
            ;2136 ;          The diagnostic is designed to handle up to 4 (four)
            ;2137 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2138 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2139 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2140 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2141 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2142 ;     first MS780-E found will have its starting address set to 0
            ;2143 ;     (zero).
            ;2144 ;     All the other MS780-E/H's found will have their starting address
            ;2145 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2146 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2147 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2148 ;     current MS780-E unit under test.
            ;2149 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2150 ;     set to maximum.
            ;2151 ;
            ;2152 ; CALLING CONSTRAINT:
            ;2153 ;
            ;2154 ; =00
            ;2155 ;
            ;2156 ; OUTPUTS:
            ;2157 ;
            ;2158 ; rc[0e] - Contains address of Configuration Register
            ;2159 ; r[0]  - Contains TR level
            ;2160 ;
            ;2161 ; SIDE EFFECTS:
            ;2162 ;
            ;2163 ; D register is destroyed.
            ;2164 ;--
            ;2165 =0
            ;2166 FIND.MS780E:
U 1068, 0000,003D,0180,0800,0000,1088  ;2167  call[sbi.init]  ; Make sure SBI is enabled
U 1069, 0003,003C,0180,0988,0000,11C7  ;2168  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11C7, 0818,0038,1980,0800,0000,11C8  ;2169  d_k[zero]  ; Clear MS780-E/H counter
U 11C8, 0000,003C,F980,3C00,0000,11C9  ;2170  id[3e]_d  ; ...
U 11C9, 0018,0038,0580,0A80,0000,11CA  ;2171  r[0]_k[.1]  ; Init TR counter
U 11CA, 0F03,003C,0180,09F0,0000,106A  ;2172  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2173      ; ...CNFG A Reg
            ;2174 =0
            ;2175 FIND.MEM.LOOP:
U 106A, 0800,003D,0180,0A00,0000,11F0  ;2176  d_r[0],call[generate.io]; Generate address of TR level
U 106B, 0001,003C,0180,09F0,0200,106C  ;2177  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 106C, 0000,003D,8980,0800,0084,7207  ;2178 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2179  d[long]_cache.p, ; Read the specified tr level
U 106D, 0000,003C,0180,C970,0000,11CB  ;2180   lc_rc[0e]  ; ...
U 11CB, 0000,003C,0180,0A78,0010,11CC  ;2181  alu_r[0f],clk.ubcc ; Check for no response
U 11CC, 0001,013C,0180,0880,0082,106E  ;2182  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 106E, 0000,003C,0180,0800,0000,11CD  ;2183 =0 j/check.adpt.code ; Check for a memory
U 106F, 0000,003C,0180,0800,0000,11EC  ;2184  j/find.mem.lp1  ; Try next tr
            ;2185 CHECK.ADPT.CODE:
```

```
U 11CD, 0000,003C,1D80,0800,1404,71CE  ;2186  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11CE, 0000,163C,0180,0800,0000,1008  ;2187  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,71CF  ;2188 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,71D0  ;2189  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11EC  ;2190  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11EC  ;2191  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,71D5  ;2192  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11EC  ;2193  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,11D9  ;2194  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,11D9  ;2195  rc[0e]_lc,j/found.ms780e ; MS780-E
           ;2196   ;
           ;2197   ; Found an MS780-A or -C. Set the starting address
           ;2198   ; to maximum.
           ;2199   ;
U 11CF, 0818,0038,5D80,0800,0000,11D1  ;2200 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11D0, 0818,0038,0580,0800,0000,11D1  ;2201 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
           ;2202 MS780.COMMON:
U 11D1, 0819,0030,7180,0800,0000,11D2  ;2203  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11D2, 0000,003C,01F8,0803,0080,D1D3  ;2204  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11D3, 0D00,003C,0180,0800,0000,11D4  ;2205  d_dal.sc  ; Put data in bits<29:14>
           ;2206  cache.p_d[long], ; Set the starting address to maximum
U 11D4, 0000,003C,0180,A800,0000,11EC  ;2207       j/find.mem.lp1 ; and continue TR scan
           ;2208   ;
           ;2209   ; Found an MA780. Set the starting address to maximum
           ;2210   ;
           ;2211 MA780.MAX:
U 11D5, 0818,0038,39F8,0803,0000,11D6  ;2212  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11D6, 0D00,003C,0180,0803,0000,11D7  ;2213  d_dal.sc,va_va+4 ; Put in proper position
U 11D7, 0000,003C,0180,0803,0000,11D8  ;2214  va_va+4   ; Point to Port Invalidate Ctrl Register
           ;2215  cache.p_d[long], ; Set starting address to maximum
U 11D8, 0000,003C,0180,A800,0000,11EC  ;2216       j/find.mem.lp1
           ;2217 ;
           ;2218 ; Found an MS780E
           ;2219 ;
           ;2220 FOUND.MS780E:
U 11D9, 0000,003C,F9F0,2C00,0000,11DA  ;2221  q_id[3e]  ; Set up to see how many MS780-E's
           ;2222  alu_q.xor.k[.3], ; Are there Maximum units?
U 11DA, 0019,2020,0D80,0800,0010,11DB  ;2223   clk.ubcc ; Check condition codes
U 11DB, 0000,013C,0180,0800,0000,1070  ;2224  z?   ; Are we at maximum units for system
U 1070, 0000,003C,0180,0800,0000,11DC  ;2225 =0 J/ms780e.tst  ; No go find how many units ther are
U 1071, 0818,0038,C180,0800,0000,1072  ;2226  d_k[.ffff]  ; Yes set address to maximum
U 1072, 0000,003D,0180,0800,0000,11F4  ;2227 =0 call[set.start.addr] ; .....
U 1073, 0000,003C,0180,0800,0000,11EC  ;2228  j/find.mem.lp1  ; Go check to see if we are done
           ;2229 ;
           ;2230 MS780E.TST:
           ;2231  alu_rc[01],  ; Check "Found MS780E Flag"
U 11DC, 0010,0038,0180,0908,0010,11DD  ;2232   clk.ubcc ; Check condition codes
U 11DD, 0810,0138,0180,0970,0000,1074  ;2233  z?,d_rc[0e]  ; Is this unit the first MS780E ?
           ;2234 =0 j/not.first.ms780e, ; No
U 1074, 0818,0038,C180,0800,0000,1078  ;2235   d_k[.ffff] ; Set starting address
U 1075, 0001,003C,0180,0988,0000,11DE  ;2236  rc[01]_d  ; Yes put base address in rc[01]
U 11DE, 0818,0038,7980,0800,0000,11DF  ;2237  d_k[.30]  ; Set up SC to point to first unit
U 11DF, 0019,0014,F580,0800,0082,11E0  ;2238  alu_d+k[.a],sc_alu ; ...
U 11E0, 0810,0038,0180,0908,0000,11E1  ;2239  d_rc[01]  ; Put base address of unit in D
U 11E1, 0000,003C,0180,3400,0000,11E2  ;2240  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11E2, 0F00,003C,0180,0800,0000,1076  ;2241  d_0   ; Prepare to set starting address to Zero
U 1076, 0000,003D,0180,0800,0000,11F4  ;2242 =0 call[set.start.addr] ; Go do it
          ;2243  j/find.mem.lp1,  ; Check to see if we are done
U 1077, 0003,003C,0180,09E8,0000,11EC  ;2244   rc[0d]_0 ; ....
          ;2245 =0
          ;2246 NOT.FIRST.MS780E:
U 1078, 0000,003D,0180,0800,0000,11F4  ;2247  call[set.start.addr] ; Go set starting address to maximum
U 1079, 0000,003C,F9F0,2C00,0000,11E3  ;2248  q_id[3e]  ; Get the number of MS780-Es found
U 11E3, 0819,2014,0580,0800,0000,11E4  ;2249  d_q+k[.1]  ; Increment this counter
U 11E4, 0000,003C,F980,3C00,0000,11E5  ;2250  id[3e]_d  ; Save the counter
U 11E5, 0018,0038,11C0,0800,0000,11E6  ;2251  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2252      ; ...were the I/O base addresses will be stored
U 11E6, 081D,2000,7980,0800,0000,11E7  ;2253  d_q-d,k[.30]  ; ... adjust pointer
U 11E7, 0819,0014,7980,0800,0000,11E8  ;2254  d_d+k[.30]  ; Point the SC to the ID register
U 11E8, 0000,003C,F580,0800,0000,11E9  ;2255  k[.a]   ; ...to receive I/O base address
U 11E9, 0019,0014,F580,0800,0082,11EA  ;2256  alu_d+k[.a],sc_alu ; ...
U 11EA, 0810,0038,0180,0970,0000,11EB  ;2257  d_rc[0e]  ; Get base address to d
U 11EB, 0000,003C,0180,3400,0000,11EC  ;2258  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2259   ;
          ;2260   ; Try next tr
          ;2261   ;
          ;2262 FIND.MEM.LP1:
U 11EC, 0818,0014,0580,0A80,0000,11ED  ;2263  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2264  alu_d.and.k[.f],
U 11ED, 0019,0034,6180,0800,0010,11EE  ;2265      clk.ubcc  ; Check if all done
U 11EE, 0000,013C,0180,0800,0000,107A  ;2266  z?   ; Branch if yes
U 107A, 0000,003C,0180,0800,0000,106A  ;2267 =0 j/find.mem.loop  ; Try next TR
U 107B, 0810,0038,0180,0908,0010,11EF  ;2268  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11EF, 0000,013C,0180,0800,0000,107C  ;2269  z?   ; Check condition codes
U 107C, 0001,003E,0180,09F0,0000,0002  ;2270 =0 return2,rc[0e]_d ; Yes MS780E was found
U 107D, 0000,003E,0180,0800,0000,0001  ;2271  return1   ; No MS780E found
          ;2272
```

```
            ;2273  .PAGE
            ;2274  .TOC "   Generate IO Address"
            ;2275 ;++
            ;2276 ; FUNCTIONAL DESCRIPTION:
            ;2277 ;
            ;2278 ; This routine is used to generate an SBI Configuration Register
            ;2279 ; address from a TR level.
            ;2280 ;
            ;2281 ; CALLING CONSTRAINT:
            ;2282 ;
            ;2283 ; =0
            ;2284 ;
            ;2285 ; INPUTS:
            ;2286 ;
            ;2287 ; D - Contains TR level
            ;2288 ;
            ;2289 ; OUTPUTS:
            ;2290 ;
            ;2291 ; D - Contains SBI IO address
            ;2292 ;--
            ;2293 GENERATE.IO:
U 11F0, 0000.003C.89F8.0800.0084.71F1  ;2294  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11F1, 0D00.003C.8D80.0800.0084.71F2  ;2295  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11F2, 0000.003C.0980.0800.0084.B1F3  ;2296  sc_sc-k[.2]  ; insert bit 29
U 11F3, 0801.001E.0180.0800.0000.0001  ;2297  d_d.orno. mask,return1 ; and return
            ;2298
```

```
            ;2299  .PAGE
            ;2300  .TOC "    Set Starting Address"
            ;2301 ;**
            ;2302 ; FUNCTIONAL DESCRIPTION:
            ;2303 ;
            ;2304 ; This routine is used to set the starting address of the
            ;2305 ; memory to the specified value.
            ;2306 ;
            ;2307 ; CALLING CONSTRAINT:
            ;2308 ;
            ;2309 ; =0
            ;2310 ;
            ;2311 ; INPUTS:
            ;2312 ;
            ;2313 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2314 ;     (B Register Image divided by 2**16)
            ;2315 ; rc[0] - Contains Address of Register A
            ;2316 ;
            ;2317 ; OUTPUTS:
            ;2318 ;
            ;2319 ; d - Contains aligned starting address (B Register Image)
            ;2320 ;
            ;2321 ; SIDE EFFECTS:
            ;2322 ;
            ;2323 ; d,q,va, and sc are destroyed
            ;2324 ;--
            ;2325 SET.START.ADDR:
U 11F4, 0010,0038,6580,0970,0284,71F5  ;2326  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11F5, 0000,003C,01F8,0803,0000,11F6  ;2327  va_va+4,q_0  ; Set address to Register B
            ;2328  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11F6, 0D00,003C,0980,0800,0084,B1F7  ;2329      sc_sc-k[.2]  ; Setup to insert bit 14
U 11F7, 0801,001C,0180,0800,0000,11F8  ;2330  d_d.ornot.mask  ; Insert Write Enable bit
U 11F8, 0000,003E,0180,A800,0000,0001  ;2331  cache.p_d[long],return1 ; Set the starting address and return
            ;2332
            ;2333
```

```
            ;2334  .PAGE
            ;2335  .TOC "   ENABLE NEXT MS780-E"
            ;2336 ;**
            ;2337 ; FUNCTIONAL DESCRIPTION:
            ;2338 ;
            ;2339 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2340 ;    They will be tested  separately, according to the TR level at which
            ;2341 ;    they are located,  starting with the one  at the lowest level first.
            ;2342 ;          When  a test has finished  with the first unit, it will have to call
            ;2343 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2344 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2345 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2346 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2347 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2348 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2349 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2350 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2351 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2352 ;
            ;2353 ; CALLING CONSTRAINT:
            ;2354 ;
            ;2355 ;  =00
            ;2356 ;
            ;2357 ;--
            ;2358 ENABLE.NEXT.MS780E:
U 11F9, 0000,003C,F9F0,2C00,0000,11FA  ;2359  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2360  ALU_Q,   ; Check to see if it is zero
U 11FA, 0001,203C,0180,0800,0010,11FB  ;2361     CLK.UBCC  ; Check Condition Codes
U 11FB, 0000,013C,0180,0800,0000,107E  ;2362  Z?   ; ...for zero
U 107E, 0000,003C,0180,0800,0000,11FC  ;2363 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 107F, 0000,003E,0180,0800,0000,0002  ;2364  RETURN2   ; Zero No more units to test
            ;2365 ENABLE.NEXT:
U 11FC, 0818,0038,C180,0800,0000,1080  ;2366  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1080, 0000,003D,0180,0800,0000,11F4  ;2367 =0 CALL[SFT.START.ADDR] ; Go set starting address of unit to maxium
U 1081, 0818,0038,7980,0800,0000,11FD  ;2368  D_K[.30]  ; Prepare to point to the ID register holding
            ;2369     ; ...the I/O Base address of the next MS780-E
U 11FD, 0819,0014,1180,0800,0000,11FE  ;2370  D_D+K[.4]  ; ...
U 11FE, 0000,003C,F580,0800,0000,11FF  ;2371  K[.A]   ; ...
U 11FF, 0819,0014,F580,0800,0000,1200  ;2372  D_D+K[.A]  ; ...
U 1200, 0000,003C,F9F0,2C00,0000,1201  ;2373  Q_ID[3E]  ; Get MS780-E Counter
            ;2374  D_D-Q,   ; D now holds the pointer to the ID register
U 1201, 081D,0000,0180,0800,0082,1202  ;2375   SC_ALU  ; ...
U 1202, 0000,003C,01F0,2400,0000,1203  ;2376  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1203, 0001,203C,0180,09F0,0000,1204  ;2377  RC[0E]_Q  ; Save it also in RC[0E]
U 1204, 0F03,003C,0180,09E8,0000,1082  ;2378  RC[0D]_0,D_0  ; Set starting address to zero
U 1082, 0000,003D,0180,0800,0000,11F4  ;2379 =0 CALL[SET.START.ADDR] ; ....
U 1083, 0F00,003C,F9F0,2C00,0000,1205  ;2380  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1205, 081D,2008,0180,0800,0000,1206  ;2381  D_Q-D-1   ; Subtract 1 from total
U 1206, 0000,003C,F980,3C00,0000,1084  ;2382  ID[3E]_D  ; Adjust the pointer
U 1084, 0000,003D,0180,0800,0000,11BF  ;2383 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 1085, 0000,003E,0180,0800,0000,0001  ;2384  RETURN1   ; Tell caller another unit was found
            ;2385
```

```
            ;2386  .PAGE
            ;2387  .TOC "    Set Trap Mask"
            ;2388 ;**
            ;2389 ; FUNCTIONAL DESCRIPTION:
            ;2390 ;
            ;2391 ; This routine is used to set a bit in the Micro Trap Mask
            ;2392 ; R[0F].
            ;2393 ;
            ;2394 ; CALLING CONSTRAINT:
            ;2395 ;
            ;2396 ; =0
            ;2397 ;
            ;2398 ; INPUTS:
            ;2399 ;
            ;2400 ; SC - Contains bit number to set.
            ;2401 ;
            ;2402 ; OUTPUTS:
            ;2403 ;
            ;2404 ; rc[0E] - Contains the current trap mask
            ;2405 ;
            ;2406 ; SIDE EFFECTS:
            ;2407 ;
            ;2408 ; d regsiter is destroyed
            ;2409 ;--
            ;2410 SET_TRAP_MASK:
U 1207, 0800,003C,0180,0A78,0000,1208  ;2411  d_r[0f]
U 1208, 0801,001C,0180,0AF8,0000,1209  ;2412  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1209, 0001,003E,0180,0AF0,0000,0001  ;2413  r[0E]_d,return1    ; Save mask and return
            ;2414
            ;2415
```

```
            ;2416  .PAGE
            ;2417  .TOC "   CHECK UTRAP"
            ;2418 ;++
            ;2419 ;FUNCTIONAL DESCRIPTION:
            ;2420 ;
            ;2421 ; This subroutine is used to check wether a Utrap occured.
            ;2422 ; It takes the contents of the Save Utrap flags register
            ;2423 ; R[0E], and compares them to the contrnts of the Utrap
            ;2424 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2425 ; according to the results of the test  (i.e., contents of
            ;2426 ; these registers are equal or not).
            ;2427 ; CALLING CONSTRAINT:
            ;2428 ;
            ;2429 ; =00
            ;2430 ;
            ;2431 ; INPUTS:
            ;2432 ;
            ;2433 ; R[0E]- Saved Utrap Mask
            ;2434 ; R[0F]- Expected Utrap Mask
            ;2435 ;
            ;2436 ;--
            ;2437 CHECK_UTRAP:
U 120A, 0800.003C.0180.0A70.0000.120B  ;2438  D_R[0E]   ; Reg D is loaded with the trap mask
U 120B, 0001.003C.0180.09C0.0000.120C  ;2439  RC[08]_D  ; Save expected Utrap flag for error logout
U 120C, 0800.003C.0180.0A78.0000.120D  ;2440  D_R[0F]   ; Get recieved Utrap flag to D reg
U 120D, 0001.003C.0180.09B8.0000.120E  ;2441  RC[07]_D  ; Save in RC[07]
            ;2442  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 120E, 000D.0020.0180.0A70.0010.120F  ;2443        CLK.UBCC
U 120F, 0003.013C.0180.0AF8.0000.1086  ;2444  Z?,R[0F]_0  ; Branch if no traps
U 1086, 0000.003E.0180.0800.0000.0002  ;2445 =0 RETURN2   ; Utrap occured
U 1087, 0000.003E.0180.0800.0000.0001  ;2446  RETURN1    ; No Utrap  return
```

```
            ;2447  .PAGE
            ;2448  .TOC "    Initialize the SBI"
            ;2449 ;++
            ;2450 ; FUNCTIONAL DESCRIPTION:
            ;2451 ;
            ;2452 ; This routine performs the following functions:
            ;2453 ;
            ;2454 ;  Executes an SBI Unjam
            ;2455 ;  Clears the SBI Fault Latch
            ;2456 ;  Clears the SBI Error Register
            ;2457 ;
            ;2458 ; CALLING CONSTRAINT:
            ;2459 ;
            ;2460 ; =0
            ;2461 ;
            ;2462 ; SIDE EFFECTS:
            ;2463 ;
            ;2464 ; D & Q Register destroyed
            ;2465 ;--
            ;2466 =0
            ;2467 SBI.INIT:
U 1088, 0000.003D,0180,0800,0000,124C  ;2468  call[sbi.unjam]  ; Unjam the SBI
U 1089, 0818.0038,C180,0800,0000,1210  ;2469  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1210, 0801.0028,6580,3C00,0000,1211  ;2470  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1211, 0F00.003C,6D80,3C00,0000,1212  ;2471  id[fault]_d,d_0
U 1212, 0000.003C,6D80,3C00,0000,1213  ;2472  id[fault]_d
U 1213, 0000.003E,6580,3C00,0000,0001  ;2473  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2474
            ;2475
```

```
            ;2476  .PAGE
            ;2477  .TOC "   Select Controller"
            ;2478 ;++
            ;2479 ; FUNCTIONAL DESCRIPTION:
            ;2480 ;
            ;2481 ; This routine is used to put the memory system into the
            ;2482 ; specified configuration with respect to controller select.
            ;2483 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2484 ; a RETURN2 is made.
            ;2485 ;
            ;2486 ; CALLING CONSTRAINT:
            ;2487 ;
            ;2488 ; =00
            ;2489 ;
            ;2490 ; INPUTS:
            ;2491 ;
            ;2492 ; d - Contains value to write to bits<2:0> of Register A
            ;2493 ; rc[0e] - Address of Register A
            ;2494 ;
            ;2495 ; SIDE EFFECTS:
            ;2496 ;
            ;2497 ; sc,d,va are destroyed
            ;2498 ;--
            ;2499 SELECT.CNTRLR:
U 1214. 0010.0038.0180.0970.0284.7215  ;2500  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1215. 0801.001C.0180.0800.0000.1216  ;2501  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1216. 0000.003C.0180.A800.0000.1217  ;2502  cache.p_d[long]  ; Write the configuration Register
U 1217. 0818.0038.5D80.0800.0000.1218  ;2503  d_k[.7]   ; Get Misconfiguration bits
U 1218. 0000.003C.61F8.0800.0084.7219  ;2504  sc_k[.F],q_0  ; ...
U 1219. 0D00.003C.0180.0800.0000.121A  ;2505  d_dal.sc  ; ... D = x38000
U 121A. 0000.003C.01E0.0800.0000.121B  ;2506  q_d   ; Save mask
U 121B. 0000.003C.0180.C800.0000.121C  ;2507  d[long]_cache.p  ; Read CNFG A Reg and
            ;2508  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 121C. 001D.2034.0180.0800.0010.121D  ;2509   clk.ubcc ; ...
U 121D. 0000.013C.0180.0800.0000.108A  ;2510  z?   ; Branch if no
U 108A. 0000.003E.0180.0800.0000.0001  ;2511 =0 RETURN1   ; Bad configuration
U 108B. 0000.003E.0180.0800.0000.0002  ;2512  RETURN2   ; Configuration ok
            ;2513
```

```
            ;2514  .PAGE
            ;2515  .TOC "   SELECT LOWER CONTROLLER"
            ;2516 ;*****************************************************************
            ;2517 ;++
            ;2518 ;
            ;2519 ; Functional Description:
            ;2520 ; ----------------------
            ;2521 ;
            ;2522 ; This subroutine can be called to select the lower
            ;2523 ; Controller on a MS780-E/H.
            ;2524 ; If the Lower controller is misconfigured then a
            ;2525 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2526 ;
            ;2527 ; Calling Constraint: =00
            ;2528 ; ------------------
            ;2529 ;
            ;2530 ;
            ;2531 ; Inputs:
            ;2532 ; ------
            ;2533 ;
            ;2534 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2535 ;
            ;2536 ; Outputs:
            ;2537 ; -------
            ;2538 ;
            ;2539 ; RC[0D] will have the address of CNFG Register C
            ;2540 ;        ( 2000x008 )
            ;2541 ;
            ;2542 ; Side Effects:
            ;2543 ; ------------
            ;2544 ;
            ;2545 ; Contents of the following registers will lost:
            ;2546 ;
            ;2547 ; D
            ;2548 ;
            ;2549 ; The Lower controller on the MS780-E/H will be configured
            ;2550 ; when the subroutine is exitted with a RETURN2
            ;2551 ;--
            ;2552 ;*****************************************************************
            ;2553 =00
            ;2554 SELECT.LOWER:
            ;2555  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,1214  ;2556   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2557  RETURN1    ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2558  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;2559  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2560   RETURN2   ; Let caller know that the controller
            ;2561       ; ...was configured properly
            ;2562
```

```
            ;2563  .PAGE
            ;2564  .TOC "   SELECT UPPER CONTROLLER"
            ;2565 ;*************************************************************
            ;2566 ;**
            ;2567 ;
            ;2568 ; Functional Description:
            ;2569 ; ----------------------
            ;2570 ;
            ;2571 ; This subroutine can be called to select the upper
            ;2572 ; Controller on a MS780-E/H.
            ;2573 ; If the Upper controller is misconfigured then a
            ;2574 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2575 ;
            ;2576 ; Calling Constraint: =00
            ;2577 ; ------------------
            ;2578 ;
            ;2579 ;
            ;2580 ; Inputs:
            ;2581 ; ------
            ;2582 ;
            ;2583 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2584 ;
            ;2585 ; Outputs:
            ;2586 ; -------
            ;2587 ;
            ;2588 ; RC[0D] will have the address of CNFG Register D
            ;2589 ;        ( 2000x010 )
            ;2590 ;
            ;2591 ; Side Effects:
            ;2592 ; ------------
            ;2593 ;
            ;2594 ; Contents of the following registers will lost:
            ;2595 ;
            ;2596 ; D
            ;2597 ;
            ;2598 ; The Upper controller on the MS780-E/H will be configured
            ;2599 ; when the subroutine is exitted with a RETURN2
            ;2600 ;--
            ;2601 ;*************************************************************
            ;2602 =00
            ;2603 SELECT.UPPER:
            ;2604  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,1214  ;2605   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2606  RETURN1     ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2607  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2608  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2609   RETURN2    ; Let caller know that the controller
            ;2610       ; ...was configured properly
            ;2611
```

```
;2612  .PAGE
;2613  .TOC "   CONFIGURE MS780-E/H"
;2614 ;*************************************************************
;2615 ;++
;2616 ;
;2617 ; Functional Description:
;2618 ; ----------------------
;2619 ;
;2620 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2621 ; specifically  check  the  memory, but  make use of it to execute.
;2622 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2623 ; that  if  a  Read/Write operation will take place no false errors
;2624 ; will be logged due to misconfigured memory.
;2625 ;
;2626 ; Calling Constraint: =00
;2627 ; ------------------
;2628 ;
;2629 ;
;2630 ; Inputs:
;2631 ; ------
;2632 ;
;2633 ; RC[0E] - I/O BASE OF MS780-E/H
;2634 ;
;2635 ;
;2636 ; Outputs:
;2637 ; -------
;2638 ;
;2639 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2640 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2641 ;      OR IF NO MS780-E/Hs WERE FOUND
;2642 ;      ON THE SBI
;2643 ;
;2644 ; Side Effects:
;2645 ; ------------
;2646 ;
;2647 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2648 ;
;2649 ;
;2650 ;
;2651 ;--
;2652 ;*************************************************************
;2653 CONFIG.MS780E:
U 121E, 0818,0038,0D80,0800,0000,121F  ;2654  D_K[.3]    ; Set up flag for the Fail Chain
U 121F, 0001,003C,0180,09E8,0000,1014  ;2655  RC[0D]_D   ; ...
U 1014, 0000,003D,0180,0800,0000,1068  ;2656 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1015, 0000,003D,0980,0800,0084,7064  ;2657  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 1016, 0000,003C,0180,0800,0000,1018  ;2658  NOP    ; OK. Found at least one MS780-E/H
;2659 =
;2660 CONFIG.RETRY:    ; Entry point for the case in which the
;2661       ; ...first MS780-E/H could not be
;2662       ; ...properly configured
U 1018, 0000,003D,0180,0800,0000,1004  ;2663 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1019, 0000,003C,0180,0800,0000,101C  ;2664  J/CNFG.LMIS   ; Lower controller misconfigured
U 101A, 0000,003E,0180,0800,0000,0001  ;2665  RETURN1    ; OK. Lower Controller is configured
;2666 =
```

```
            ;2667 =00
            ;2668 CNFG.LMIS:
U 101C, 0000,003D,0180,0800,0000,1010  ;2669  CALL[SELECT.UPPER]  ; Select the upper controller
U 101D, 0000,003C,0180,0800,0000,1020  ;2670  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 101E, 0000,003E,0180,0800,0000,0001  ;2671  RETURN1    ; OK. Upper Controller is configured
            ;2672 =
            ;2673 =00
            ;2674 CNFG.UMIS:
U 1020, 0000,003D,0180,0800,0000,11F9  ;2675  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2676       ; ...MS780-E/H found so go and see
            ;2677       ; ...if there are any more
U 1021, 0000,003C,0180,0800,0000,1220  ;2678  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2679       ; ...try to configure it
U 1022, 0818,0038,0D80,0800,0000,1023  ;2680  D_K[.3]    ; Set Flag for the Fail Chain
U 1023, 0001,003C,0180,09E8,0000,1220  ;2681  RC[0D]_D
U 1220, 0000,003D,0980,0800,0084,7064  ;2682  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2683       ; ...abort the test
            ;2684 =
            ;2685
```

```
            ;2686  .PAGE
            ;2687  .TOC "   ENBFLT ROUTINE"
            ;2688 ;*****************************************************************
            ;2689 ;*
            ;2690 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2691 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2692 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2693 ;
            ;2694 ; D AND SC GET CLOBBERED
            ;2695 ;-
            ;2696 ;*****************************************************************
U 1221. 0F00.003C.2180.0800.0084.7222  ;2697 ENBFLT: SC_K[.14],D_0
U 1222. 0000.003C.0580.0800.0084.B223  ;2698  SC_SC-K[.1]
U 1223. 0801.001C.0180.0800.0000.1224  ;2699  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 1224. 0000.003C.6D80.3C00.0000.1225  ;2700  ID[FAULT]_D
U 1225. 0600.003C.0180.0800.0000.1226  ;2701  D_D.RIGHT.SI/ZERO
U 1226. 0000.003C.6D80.3C00.0000.1227  ;2702  ID[FAULT]_D
U 1227. 0000.003E.0180.0800.0000.0001  ;2703  RETURN1
            ;2704
```

```
            ;2705  .PAGE
            ;2706  .TOC "   NOINTR ROUTINE"
            ;2707 ;*****************************************************************
            ;2708 ;+
            ;2709 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
            ;2710 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
            ;2711 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
            ;2712 ; IF ANY INTERRUPTS OCCURRED:
            ;2713 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
            ;2714 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
            ;2715 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
            ;2716 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
            ;2717 ;  RETURN1 IS USED TO GET BACK
            ;2718 ; ELSE
            ;2719 ;  RETURN2 IS USED
            ;2720 ;
            ;2721 ; THE D AND Q REGISTERS GET CLOBBERED.
            ;2722 ;
            ;2723 ;-
            ;2724 ;*****************************************************************
            ;2725 =0
U 108C, 0000,003D,0180,0800,0000,122D  ;2726 NOINTR: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 108D, 0000,003C,35F0,2C00,0000,1228  ;2727  Q_ID[VECTOR]
U 1228, 0019,2034,8180,0800,0010,1229  ;2728  ALU_Q.AND.K[.3FF],CLK.UBCC
U 1229, 0000,013C,0180,0800,0000,108E  ;2729  Z?
U 108E, 0000,003C,0180,0800,0000,1231  ;2730 =0 J/ERRFLT   ; INTR. OCCURRED,SET FLAG,GET DATA
U 108F, 0000,003E,0180,0800,0000,0002  ;2731  RETURN2    ; NO INTR, RETURN2
            ;2732 =0
U 1090, 0000,003D,0180,0800,0000,122D  ;2733 WRTOUT: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1091, 0000,003C,35F0,2C00,0000,122A  ;2734  Q_ID[VECTOR]
U 122A, 0819,2034,8180,0800,0000,122B  ;2735  ALU_Q.AND.K[.3FF],D_ALU
U 122B, 0019,0020,A580,0800,0010,122C  ;2736  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 122C, 0000,013C,0180,0800,0000,1092  ;2737  Z?
U 1092, 0000,003C,0180,0800,0000,1231  ;2738 =0 J/ERRFLT   ; WRONG VECTOR,SET FLAG,GET DATA
U 1093, 0000,003E,0180,0800,0000,0002  ;2739  RETURN2    ; OK, RETURN2
U 122D, 0F00,003C,0180,0800,0000,122E  ;2740 STROBE: D_0    ; SET LOWEST IPL ACTIVE
U 122E, 0000,003C,3D80,3C00,0000,122F  ;2741  ID[PSL]_D   ;
U 122F, 0000,003C,0180,0800,4000,1230  ;2742  IEK/ISTR   ; STROBE INTERRUPT
U 1230, 0000,003E,0180,0800,0000,0001  ;2743  RETURN1    ; WAIT BEFORE LATCHING VECTOR REGISTER
U 1231, 0018,0038,0580,09A8,0000,1232  ;2744 ERRFLT: RC[5]_K[.1]   ; Set Flag
U 1232, 0001,203C,65F0,2DC0,0000,1233  ;2745  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 1233, 0001,203C,6DF0,2DB8,0000,1234  ;2746  RC[7]_Q,Q_ID[FAULT]  ; RC[7]_SBI.ERR
U 1234, 0001,203E,0180,09B0,0000,0001  ;2747  RC[6]_Q,RETURN1   ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
            ;2748
            ;2749 ;/////////////////////////////////////////////////////////////////
            ;2750 ;+
            ;2751 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2752 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2753 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2754 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2755 ;
            ;2756 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2757 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2758 ;
            ;2759 ;  =0 D_0,CALL,J/DC5
```

```
            ;2760 ;   NOP
            ;2761 ;  =0 CALL,J/DC5
            ;2762 ;-
            ;2763
U 1235, 0018,0038,1980,0800,0000,1245  ;2764 DC0: ALU_0(K),J/S00
U 1236, 0018,0038,1980,0800,0000,1246  ;2765 DC1: ALU_0(K),J/S01
U 1237, 0018,0038,1980,0800,0000,1247  ;2766 DC2: ALU_0(K),J/S10
U 1238, 0018,0038,1980,0800,0000,1248  ;2767 DC3: ALU_0(K),J/S11
U 1239, 0018,0038,0980,0800,0000,1245  ;2768 DC4: ALU_K[.2],J/S00
U 123A, 0018,0038,0980,0800,0000,1246  ;2769 DC5: ALU_K[.2],J/S01
U 123B, 0018,0038,0980,0800,0000,1247  ;2770 DC6: ALU_K[.2],J/S10
U 123C, 0018,0038,0980,0800,0000,1248  ;2771 DC7: ALU_K[.2],J/S11
U 123D, 0018,0038,0580,0800,0000,1245  ;2772 DC8: ALU_K[.1],J/S00
U 123E, 0018,0038,0580,0800,0000,1246  ;2773 DC9: ALU_K[.1],J/S01
U 123F, 0018,0038,0580,0800,0000,1247  ;2774 DCA: ALU_K[.1],J/S10
U 1240, 0018,0038,0580,0800,0000,1248  ;2775 DCB: ALU_K[.1],J/S11
U 1241, 0018,0038,C180,0800,0000,1245  ;2776 DCC: ALU_K[.FFFF],J/S00
U 1242, 0018,0038,C180,0800,0000,1246  ;2777 DCD: ALU_K[.FFFF],J/S01
U 1243, 0018,0038,C180,0800,0000,1247  ;2778 DCE: ALU_K[.FFFF],J/S10
U 1244, 0018,0038,C180,0800,0000,1248  ;2779 DCF: ALU_K[.FFFF],J/S11
            ;2780
U 1245, 0118,0038,1B80,0800,0000,1249  ;2781 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1246, 0118,0038,0B80,0800,0000,1249  ;2782 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 1247, 0118,0038,0780,0800,0000,1249  ;2783 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 1248, 0118,0038,C380,0800,0000,1249  ;2784 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
U 1249, 0100,003E,0380,0800,0000,0001  ;2785 SX: D_D.LEFT2,SI/7,RETURN1 ; EXIT
            ;2786
            ;2787
```

```
            ;2788  .PAGE
            ;2789  .TOC "    DELAY ROUTINE"
U 124A, 0818,0038,0180,0800,0000,105E  ;2790 DELAY: D_K[.8],J/CALLCON
            ;2791
            ;2792
            ;2793
```

```
            ;2794  .PAGE
            ;2795  .TOC "   Wait Loop Routine"
            ;2796 ;++
            ;2797 ; FUNCTIONAL DESCRIPTION:
            ;2798 ;
            ;2799 ; This routine is used to wait the specified number of machine cycles.
            ;2800 ; This routine is typically called to wait for an SBI write to
            ;2801 ; I/O space to complete.
            ;2802 ;
            ;2803 ; CALLING CONSTRAINT:
            ;2804 ;
            ;2805 ; =0
            ;2806 ;
            ;2807 ; INPUTS:
            ;2808 ;
            ;2809 ; d - Contains number of cycles to wait
            ;2810 ; alu.z condition code must not be set
            ;2811 ;
            ;2812 ; SIDE EFFECTS:
            ;2813 ;
            ;2814 ; d is destroyed
            ;2815 ;--
            ;2816 WAIT.LOOP:
U 124B, 0018,0038,6180,0800,0010,1094  ;2817  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2818      ; ...is not set
            ;2819 =0
            ;2820 W.LOOP:
            ;2821  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1094, 0819,0100,0580,0800,0010,1094  ;2822       j/W.LOOP
U 1095, 0000,003E,0180,0800,0000,0001  ;2823  return1   ; exit
            ;2824
            ;2825
```

```
            ;2826  .PAGE
            ;2827  .TOC "   SBI UNJAM (SBI CLEAR) ROUTINE"
            ;2828 ;=================================================================
            ;2829 ;/////////////////////////////////////////////////////////////////
            ;2830 ;+
            ;2831 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2832 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2833 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2834 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2835 ;-
            ;2836 CLRSBI:
            ;2837 SBI.UNJAM:
U 124C. 0818,0038,6580,8000,0010,1096  ;2838 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2839 =0
U 1096. 0819,0100,0580,8000,0010,1096  ;2840 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 1097. 0818,0038,6580,8800,0010,1098  ;2841  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2842 =0
U 1098. 0819,0100,0580,8800,0010,1098  ;2843 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 1099. 0818,0038,6580,8000,0010,109A  ;2844  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2845 =0
U 109A. 0819,0100,0580,8000,0010,109A  ;2846 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 109B. 0000,003E,0180,0800,0000,0001  ;2847  RETURN1   ; DONE.
```

```
            ;2848   .PAGE
            ;2849   .TOC "    INIT ROUTINE"
            ;2850 INIT: ;=================================================================
            ;2851   ;
            ;2852   ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2853   ;    PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2854   ;     FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2855   ;
            ;2856   ;=================================================================
            ;2857   ;
U 124D. 0003.003C.0180.0AF8.0000.109C   ;2858   R[0F]_0     ;CLEAR U-TRAPS
U 109C. 0000.003D.0180.0800.0000.124C   ;2859  =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 109D. 0000.003C.0180.0800.0000.109E   ;2860   NOP
U 109E. 0000.003D.0180.0800.0000.125D   ;2861  =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2862           ;  INTRUPT REG
U 109F. 0000.003C.0180.0800.0000.10A0   ;2863   NOP
U 10A0. 0000.003D.0180.0800.0000.1264   ;2864  =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 10A1. 0000.003C.0180.0800.0000.10A2   ;2865   NOP
U 10A2. 0000.003D.0180.0800.0000.1268   ;2866  =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 10A3. 0000.003C.0180.0800.0000.10A4   ;2867   NOP
U 10A4. 0000.003D.0180.0800.0000.1260   ;2868  =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 10A5. 0000.003C.0180.0800.0000.10A6   ;2869   NOP
U 10A6. 0000.003D.0180.0800.0000.126C   ;2870  =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 10A7. 0000.003C.0180.0800.0000.10A8   ;2871   NOP
U 10A8. 0000.003D.0180.0800.0000.1221   ;2872  =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 10A9. 0818.0038.4580.0800.0000.10AA   ;2873   D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;2874  =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 10AA. 0000.003D.7980.3C00.0000.11B6   ;2875    ID[PARITY]_D
U 10AB. 0F00.003C.0180.0800.0000.124E   ;2876   D_0
U 124E. 0000.003C.4980.3C00.0000.124F   ;2877   ID[TBER0]_D    ;SHUT TB OFF
U 124F. 0000.003E.7180.3C00.0000.0001   ;2878   ID[COMP]_D,RETURN1   ;DISABLE SILO INTERRUPTS
            ;2879   ;
            ;2880   ;=================================================================
            ;2881   ;*
            ;2882   ;CHECK FOR INTERRUPTS PENDING
            ;2883   ;
            ;2884   ; INTERRUPT  VECTOR CHECKED
            ;2885   ; ------------------------------------
            ;2886   ; WRITE TIMEOUT  ^X60
            ;2887   ; SBI FAULT  ^X5C
            ;2888   ; CRD,RDS TAG  ^X54
            ;2889   ; SILO    ^X50
            ;2890   ;
            ;2891   ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2892   ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2893   ;
            ;2894   ; DATA: EXPECTED VECTOR
            ;2895   ;  VECTOR STROBED
            ;2896   ;-
            ;2897   ;=================================================================
            ;2898   ;
            ;2899 SBIFLT: ;================================================================
            ;2900   ;CHECKS FOR THE SBI FAULT INTERRUPT, ^X5C
            ;2901   ;=================================================================
            ;2902   ;
```

```
U 1250. 0000.003C.0180.0800.4000.1251  ;2903  IEK/ISTR   ;STROBE THE VECTOR
U 1251. 0818.0038.3580.0800.0000.1252  ;2904  D_K[.50]
U 1252. 0819.0030.8580.0800.0000.1253  ;2905  D_D.OR.K[.C]   ;GENERATE THE EXPECTED VECTOR
U 1253. 0000.003C.35F0.2C00.0000.1254  ;2906  Q_ID[VECTOR]
U 1254. 0019.2034.81C0.0800.0000.1255  ;2907  Q_Q.AND.K[.3FF]
U 1255. 001D.0020.0180.0800.0010.1256  ;2908  ALU_D[XOR]Q,CLK.UBCC
U 1256. 0000.013C.0180.0800.0000.10AC  ;2909  Z?
U 10AC. 0000.003C.0180.0800.0000.1231  ;2910 =0 J/ERRFLT   ;NOT THE EXPECTED INTERRUPT
U 10AD. 0000.003E.0180.0800.0000.0002  ;2911  RETURN2
          ;2912 SILINT: ;=================================================================
          ;2913  ;CHECK FOR SILO INTERRUPT VECTOR   ^X50
          ;2914  ;=================================================================
          ;2915  ;
U 1257. 0000.003C.0180.0800.4000.1258  ;2916  IEK/ISTR   ;STROBE THE VECTOR
U 1258. 0818.0038.3580.0800.0000.1259  ;2917  D_K[.50]
U 1259. 0000.003C.35F0.2C00.0000.125A  ;2918  Q_ID[VECTOR]
U 125A. 0019.2034.81C0.0800.0000.125B  ;2919  Q_Q.AND.K[.3FF]   ;MASK
U 125B. 001D.0020.0180.0800.0010.125C  ;2920  ALU_D[XOR]Q,CLK.UBCC  ;COMPARE
U 125C. 0000.013C.0180.0800.0000.10AE  ;2921  Z?
U 10AE. 0000.003C.0180.0800.0000.1231  ;2922 =0 J/ERRFLT   ;WRONG VECTOR
U 10AF. 0000.003E.0180.0800.0000.0002  ;2923  RETURN2    ;O.K.,NORMAL RETURN
          ;2924 SETPSL: ;=================================================================
          ;2925  ;
          ;2926  ;DROP THE CPU IPR LEVEL TO
          ;2927  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
          ;2928  ; PENDING
          ;2929  ;
          ;2930  ;=================================================================
          ;2931  ;
U 125D. 0F00.003C.0180.0800.0000.125E  ;2932  D_0
U 125E. 0000.003C.3980.3C00.0000.125F  ;2933  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 125F. 0000.003E.3D80.3C00.0000.0001  ;2934  RETURN1,ID[PSL]_D
          ;2935 CLRTIM: ;=================================================================
          ;2936  ;
          ;2937  ;CLEAR THE CP TIMEOUT BIT IN THE
          ;2938  ; SBI ERROR REGISTER
          ;2939  ;
          ;2940  ;=================================================================
          ;2941  ;
U 1260. 0818.0038.0580.0800.0000.1261  ;2942  D_K[.1]
U 1261. 0000.003C.85F8.0800.0084.7262  ;2943  Q_0,SC_K[.C]
U 1262. 0D00.003C.0180.0800.0000.1263  ;2944  D_DAL.SC
U 1263. 0000.003E.6580.3C00.0000.0001  ;2945  ID[SBI.ERR]_D,RETURN1
          ;2946 CLRFLT: ;=================================================================
          ;2947  ;
          ;2948  ;CLEAR THE CP FAULT BIT IN THE
          ;2949  ; FAULT REGISTER
          ;2950  ;
          ;2951  ;=================================================================
          ;2952  ;
U 1264. 0818.0038.0180.0800.0000.1265  ;2953  D_K[.8]
U 1265. 0000.003C.65F8.0800.0084.7266  ;2954  Q_0,SC_K[.10]
U 1266. 0D00.003C.0180.0800.0000.1267  ;2955  D_DAL.SC
U 1267. 0000.003E.6D80.3C00.0000.0001  ;2956  ID[FAULT]_D,RETURN1
          ;2957 CLRECC: ;=================================================================
```

```
            ;2958  ;
            ;2959  ;CLEAR CRD.RDS BIT IN THE
            ;2960  ;SBI ERROR REGISTER
            ;2961  ;
            ;2962  ;=====================================================================
            ;2963  ;
U 1268. 0818.0038.0D80.0800.0000.1269  ;2964  D_K[.3]
U 1269. 0000.003C.89F8.0800.0084.726A  ;2965  Q_0,SC_K[.D]
U 126A. 0D00.003C.0180.0800.0000.126B  ;2966  D_DAL.SC
U 126B. 0000.003E.6580.3C00.0000.0001  ;2967  ID[SBI.ERR]_D,RETURN1
            ;2968 ENBECC: ;=====================================================================
            ;2969  ;
            ;2970  ;ENABLE CRD.RDS INTERRUPTS
            ;2971  ;
            ;2972  ;=====================================================================
            ;2973  ;
U 126C. 0818.0038.0580.0800.0000.126D  ;2974  D_K[.1]
U 126D. 0000.003C.61F8.0800.0084.726E  ;2975  Q_0,SC_K[.F]
U 126E. 0D00.003C.0180.0800.0000.126F  ;2976  D_DAL.SC
U 126F. 0000.003E.6580.3C00.0000.0001  ;2977  ID[SBI.ERR]_D,RETURN1
            ;2978
```

```
;2979 .PAGE "TEST 1A1 MISCELLANEOUS SILO AND SBI (CPU) INTERFACE TESTS"
;2980 ;*****************************************************************************
;2981 ;**
;2982 ; TEST   MISCELLANEOUS SILO AND SBI (CPU) INTERFACE TESTS
;2983 ;
;2984 ;
;2985 ; TEST DESCRIPTION
;2986 ;
;2987 ; SUBTEST 1 GENERATE CONSTANTS, FIND CPU ID
;2988 ;
;2989 ;
;2990 ; SUBTEST 2 TEST SETTING AND CLEARING OF THE IMPLICT
;2991 ;  INTERLOCK FLOP ON THE SBI INTERFACE. THE TEST CHECKS
;2992 ;  THAT THE WRITE OPERATION FOLLOWING THE
;2993 ;  READ.V.WCHK TO I/O SPACE IS TRANSFORMED INTO AN
;2994 ;  WRITE COMMAND. ALSO CHECKED IS THAT THE SECOND WRITE
;2995 ;  TO FOLLOW THE READ.V.WCHK (TO I/O SPACE) IS NOT
;2996 ;  TRANSFORMED TO AN INTERLOCK WRITE OPERATION( THUS
;2997 ;  SEEING THE IMPLICT INTERLOCK FLOP CLEARED).
;2998 ;
;2999 ; SUBTEST 3 THIS TEST CHECK THAT IF THE READ.V.WCHK DOES NOT RETURN
;3000 ;  GOOD DATA, THE IMPLICT INTERLOCK FLOP DOES
;3001 ;  NOT GET SET. A ADDRESS CAUSING A TIMEOUT IS USED WITH
;3002 ;  THE READ.V.WCHK OPERATION TO PREVENT GOOD DATA FROM
;3003 ;  BEING RETURNED.
;3004 ;
;3005 ; SUBTEST 4 IN THIS TEST THE SILO WILL BE USED TO VERIFY THE CPU-
;3006 ;  RETRIES THE OPERATION WHEN A NO-RESPONSE IS RETURNED
;3007 ;  BY THE  NEXUS( MEMORY). A READ OPERATION TO AN NON-
;3008 ;  EXISTANT I/O ADDRESS WILL BE TRIED AND AFTER THE SILO
;3009 ;  LOCKS, 3 RETRY SEQUENCES WILL BE READ AND
;3010 ;  CHECKED FOR IN THE SILO.
;3011 ;
;3012 ; SUBTEST 5 UNEXPECTED READ FAULT WILL BE CHECKED. THE CPU I.D.
;3013 ;  WILL BE USED TO SEE IF THE CPU CAN DETECT UNEXPECTED
;3014 ;  READ DATA.
;3015 ;
;3016 ; SUBTEST 6 ERROR COMFRIMATION ON BYTE WRITE TO I/O SPACE
;3017 ;
;3018 ; SUBTEST 7 ERROR COMFRIMATION ON EXTENTED WRITES TO I/O SPACE
;3019 ;
;3020 ; SUBTEST 8 CPU SBI-MULTIPLE-ERROR DETECTION
;3021 ;
;3022 ; SU9TEST A TESTING THE SBI-NOT-BUSY BIT
;3023 ;
;3024 ; SUATEST B STALL ON BACK TO BACK WRITES
;3025 ;
;3026 ; LOGIC DESCRIPTION
;3027 ;
;3028 ; SUBTESTS 1 - 5
;3029 ; ---------------
;3030 ;    SILO LOGIC
;3031 ;    MCT DECODE ROMS
;3032 ;    IMPLIT FLIP/FLOP LOGIC
;3033 ;    MCT CONTROL LOGIC
```

```
;3034 ;    SBI FAULT LOGIC(UNEXPECTED READ DATA )
;3035 ;
;3036 ; ERROR LOGOUT
;3037 ;
;3038 ;
;3039 ; SEE SUBTEST FOR DESCRIPTION
;3040 ;
;3041 ;--
;3042 ;================================================================================
;3043 ;   SCRATCH PADS
;3044 ; RA       II IIII = CPU I.D.
;3045 ; 0 C/A INTERLOCK WR  00II III0 1101 1100  0-0-0-0
;3046 ; 2 C/A EXT READ   00II III0 1110 0000  0-0-0-0
;3047 ; 3 C/A MASKED READ   00II III0 1100 0100  0-0-0-0
;3048 ; 4 C/A MASKED WRITE  00II III0 1100 1000  0-0-0-0
;3049 ; 9 INTERLOCK MASK( SILO )  0100 0000 0000 0000  0-0-0-0
;3050 ; A CPU ARB (ARBITRITION)       0-0-0-0  <-TR->
;3051 ; B UNCONDITIONAL LOCK OFF 16 TIC   ^X 6000 0000
;3052 ; E
;3053 ; F U-TRAP FLAG
;3054 ;
;3055 ;
;3056 ; RC
;3057 ; 0 CONFIG REG A ADDRESS    ^X 2000 2000
;3058 ; 1 CONFIG REG B ADDRESS    ^X 2000 2004
;3059 ; 2 CONFIG REG C ADDRESS    ^X 2000 2008
;3060 ; 3 CPU I.D.
;3061 ; 4 CPU ID SHIFTED TO MAINT ID POSITION
;3062 ; A ADDRESS FLAG
;3063 ; B OPERATION POINTER
;3064 ; C SILO COUNT
;3065 ; D DATA FROM SILO
;3066 ; E EXPECTED DATA
;3067 ; F # ARG TO CONSOLE
;3068 ;
;3069 ;================================================================================
;3070 ;
;3071 ;
;3072 ;
;3073 =0
U 10B0, 0000,003D,0180,0800,0000,11A1  ;3074 S05000: NEWTST
U 10B1, 0000,003C,0180,0800,0000,10B2  ;3075  NOP
U 10B2, 0000,003D,0180,0800,0000,121E  ;3076 =0 CALL[CONFIG.MS780E]  ; Find MS780E on the SBI and
        ;3077       ; ...configure it properly
U 10B3, 0F00,003C,ED80,0800,0084,7270  ;3078  D_0,SC_K[.1B]   ;RA 9
U 1270, 0000,003C,0D80,0800,0084,9271  ;3079  SC_SC+K[.3]
U 1271, 0801,001C,0180,0AC8,0000,1272  ;3080  R[9]_ALU,D_D.ORNOT.MASK
U 1272, 0000,003C,01E0,0800,0000,1273  ;3081  Q_D     ;RA B
U 1273, 0600,003C,0180,0800,0000,1274  ;3082  D_D.RIGHT
U 1274, 081D,0030,0180,0AD8,0000,1275  ;3083  R[0B]_ALU,D_D.OR.Q
U 1275, 0018,0038,F580,09F8,0000,1276  ;3084  RC[0F]_K[.A]
        ;3085    ;=============================================================
        ;3086    ; FIND CPU ID
        ;3087    ;=============================================================
U 1276, 0003,003C,0180,0990,0000,10B4  ;3088  RC[2]_0    ;SET INTIAL CPU ID
```

```
U 10B4, 0000,003D,0180,0800,0000,11BA  ;3089 =0 ERLOOP
U 10B5, 0000,003C,0180,0800,0000,10B6  ;3090 S05010: NOP
U 10B6, 0000,003D,0180,0800,0000,124D  ;3091 =0 CALL[INIT]
U 10B7, 0F00,003C,7D80,0800,0084,7277  ;3092  SC_K[.18],D_0    ;GET FORCE MX BIT
U 1277, 0000,003C,1180,0800,0084,9278  ;3093  SC_SC+K[.4]
U 1278, 0801,001C,0180,0800,0000,1279  ;3094  D_D.ORNOT.MASK
U 1279, 0000,003C,75F0,2C00,0000,127A  ;3095  Q_ID[MAINT]    ;GET CACHE OFF BITS
U 127A, 081D,0030,0180,0800,0000,127B  ;3096  D_D[OR]Q    ;OR THE TWO TOGETHER
U 127B, 0000,003C,7580,3C00,0000,127C  ;3097  ID[MAINT]_D    ;SAVE IT IN DESTINATION
U 127C, 0810,0038,0180,0910,0000,127D  ;3098  D_RC[2]
U 127D, 0019,0020,6580,0800,0010,127E  ;3099  ALU_D[XOR]K[.10],CLK.UBCC
U 127E, 0819,0114,0580,0990,0000,10B8  ;3100  Z?,RC[2]_ALU,D_D+K[.1]  ;GET NEXT ID
U 10B8, 0000,003C,0180,0800,0000,127F  ;3101 =0 J/S05020
U 10B9, 0000,003D,0180,0800,0000,11BB  ;3102  ERROR1     ;DIDN'T SEE A MULTIPLE XMITTER
        ;3103       ;AFTER ALL 16 ID'S CHECKED
U 127F, 0000,003C,21F8,0800,0084,7280  ;3104 S05020: SC_K[.14],Q_0    ;SHIFT TEST ID TO MAINT
U 1280, 0000,003C,0D80,0800,0084,9281  ;3105  SC_SC+K[.3]    ; ID POSITION
U 1281, 0D00,003C,0180,0800,0000,1282  ;3106  D_DAL.SC
U 1282, 0000,003C,75F0,2C00,0000,1283  ;3107  Q_ID[MAINT]
U 1283, 081D,0030,0180,0800,0000,1284  ;3108  D_D[OR]Q    ;GOT FORCE MX,ID AND
U 1284, 0000,003C,7580,3C00,0000,1285  ;3109  ID[MAINT]_D    ;  CACHE OFF
U 1285, 0018,0038,1980,0800,0200,1286  ;3110  VA_K[ZERO]
U 1286, 0000,003C,0180,A800,0000,10BA  ;3111  MCT/WRITE.P    ;WRITE CAUSING MULTIPLE
        ;3112 =0 D_K[.6],    ; XMITTER
U 10BA, 0818,0039,D580,0800,0000,124B  ;3113   CALL[WAIT.LOOP]
U 10BB, 0F00,003C,0180,0800,0000,1287  ;3114  D_0
U 1287, 0000,003C,7580,3C00,0000,1024  ;3115  ID[MAINT]_D    ;CLEAR FORCE FAULT
U 1024, 0000,003D,0180,0800,0000,1250  ;3116 =00 CALL[SBIFLT]
U 1025, 0000,003C,0180,0800,0000,1292  ;3117  J/S05030    ;NO SBI FAULT,GOT ID
U 1026, 0818,0038,85F8,0800,0000,1027  ;3118  D_K[.C],Q_0    ;CHECK THE FAULT BITS SET
U 1027, 0000,003C,7D80,0800,0084,7288  ;3119  SC_K[.18]
U 1288, 0D00,003C,0180,0800,0000,1289  ;3120  D_DAL.SC
U 1289, 0001,003C,0180,09E0,0000,128A  ;3121  RC[9C]_D    ;SAVE EXPECTED DATA
U 128A, 0818,0038,F1F8,0800,0000,128B  ;3122  D_K[.FFFC],Q_0    ;GENERATE MASK FOR SBI FAULT BITS
U 128B, 0000,003C,2180,0800,0084,728C  ;3123  SC_K[.14]
U 128C, 0000,003C,1180,0800,0084,928D  ;3124  SC_SC+K[.4]
U 128D, 0D00,003C,0180,0800,0000,128E  ;3125  D_DAL.SC
U 128E, 0000,003C,6DF0,2C00,0000,128F  ;3126  Q_ID[FAULT]    ;GET FAULT REGISTER IMAGE
U 128F, 081D,0034,0180,09D8,0000,1290  ;3127  RC[0B]_ALU,D_ALU,ALU_D[AND]Q ;MASK AND SAVE
U 1290, 0011,0020,0180,0960,0010,1291  ;3128  ALU_D[XOR]RC[0C],CLK.UBCC ;ARE THE RIGHT BITS SET
U 1291, 0000,013C,0180,0800,0000,10BC  ;3129  Z?
U 10BC, 0000,003D,0180,0800,0000,11BB  ;3130 =0 ERROR1     ;WRONG FAULT BITS SET
U 10BD, 0000,003C,0180,0800,0000,10B5  ;3131  J/S05010    ;GO TRY NEXT ID
U 1292, 0810,0038,01F8,0910,0000,1293  ;3132 S05030: D_RC[2],Q_0    ;SHIFT ID TO SBI SILO ID
U 1293, 0000,003C,7D80,0800,0084,7294  ;3133  SC_K[.18]    ; POSITION
U 1294, 0000,003C,0580,0800,0084,9295  ;3134  SC_SC+K[.1]
U 1295, 0D00,003C,0180,0800,0000,1296  ;3135  D_DAL.SC
U 1296, 0001,003C,0180,0998,0000,1297  ;3136  RC[3]_D    ;SAVE IT IN RC 3
U 1297, 0818,0038,89F8,0800,0000,1298  ;3137  D_K[.D],Q_0    ;RA 0
U 1298, 0000,003C,1180,0800,0084,7299  ;3138  SC_K[.4]
U 1299, 0D00,003C,0180,0800,0000,129A  ;3139  D_DAL.SC
U 129A, 0819,0030,8580,0800,0000,129B  ;3140  D_D.OR.K[.C]
U 129B, 0000,003C,6580,0800,0084,729C  ;3141  SC_K[.10]
U 129C, 0D00,003C,0180,0800,0000,129D  ;3142  D_DAL.SC
U 129D, 0811,0030,0180,0918,0000,129E  ;3143  D_ALU,ALU_D[OR]RC[3]
```

```
U 129E. 0001.003C.0180.0A80.0000.10BE  ;3144  R[0]_D
U 10BE. 0000.003D.0180.0800.0000.1243  ;3145 =0 CALL[DCE]   ;RA 2
U 10BF. 0000.003C.6580.0800.0084.729F  ;3146  SC_K[.10]
U 129F. 0000.003C.1180.0800.0084.92A0  ;3147  SC_SC+K[.4]
U 12A0. 0D00.003C.0180.0800.0000.12A1  ;3148  D_DAL.SC
U 12A1. 0811.0030.0180.0918.0000.12A2  ;3149  D_ALU,ALU_D[OR]RC[3]
U 12A2. 0001.003C.0180.0A90.0000.12A3  ;3150  R[2]_D
U 12A3. 0818.0038.85F8.0800.0000.12A4  ;3151  D_K[.C],Q_0   ;RA 3
U 12A4. 0000.003C.1180.0800.0084.72A5  ;3152  SC_K[.4]
U 12A5. 0D00.003C.0180.0800.0000.12A6  ;3153  D_DAL.SC
U 12A6. 0819.0030.1180.0800.0000.12A7  ;3154  D_D.OR.K[.4]
U 12A7. 0000.003C.6580.0800.0084.72A8  ;3155  SC_K[.10]
U 12A8. 0D00.003C.0180.0800.0000.12A9  ;3156  D_DAL.SC
U 12A9. 0811.0030.0180.0918.0000.12AB  ;3157  D_ALU,ALU_D[OR]RC[3]
U 12AB. 0001.003C.0180.0A98.0000.12AC  ;3158  R[3]_D
U 12AC. 0818.0038.85F8.0800.0000.12AD  ;3159  D_K[.C],Q_0   ;RA 4
U 12AD. 0000.003C.1180.0800.0084.72AE  ;3160  SC_K[.4]
U 12AE. 0D00.003C.0180.0800.0000.12AF  ;3161  D_DAL.SC
U 12AF. 0819.0030.0180.0800.0000.12B0  ;3162  D_D.OR.K[.8]
U 12B0. 0000.003C.6580.0800.0084.72B1  ;3163  SC_K[.10]
U 12B1. 0D00.003C.0180.0800.0000.12B2  ;3164  D_DAL.SC
U 12B2. 0811.0030.0180.0918.0000.12B3  ;3165  D_ALU,ALU_D[OR]RC[3]
U 12B3. 0001.003C.0180.0AA0.0000.12B4  ;3166  R[4]_D
U 12B4. 0F10.0038.0180.0910.0082.12B5  ;3167  SC_ALU,ALU_RC[2],D_0  ;RA A
U 12B5. 0801.001C.0180.0AD0.0000.10C0  ;3168  R[0A]_ALU,D_D.ORNOT.MASK
U 10C0. 0000.003D.0180.0800.0000.11C0  ;3169 =0 SUBTEST
U 10C1. 0000.003C.0180.0800.0000.10C2  ;3170  NOP
U 10C2. 0000.003D.0180.0800.0000.11C2  ;3171 =0 CALL[SUBTEST1]
U 10C3. 0500.003C.0180.0800.0000.12B6  ;3172  D_D.LEFT
U 12B6. 0001.003C.0180.0AE8.0000.10C4  ;3173  R[0D]_D
          ;3174 ;
          ;3175 ;================================================================================
          ;3176 ;*
          ;3177 ; SUBTEST 2 IMPLICT INTERLOCK OPERATIONS TO I/O SPACE
          ;3178 ;
          ;3179 ;
          ;3180 ;  THIS SUBTEST CHECKS THAT THE IMPLICT INTERLOCK
          ;3181 ;  FLIP/FLOP ON THE SBI(CPU) INTERFACE IS SET
          ;3182 ;  AND CLEARED PROPERLY.  CHECK TO SEE THE WRITE
          ;3183 ;  OPERATION FOLLOWING THE READ.V.WCHK TO
          ;3184 ;  I/O SPACE TRANSFORMED TO AN INTERLOCK
          ;3185 ;  WRITE OPERATION.
          ;3186 ;
          ;3187 ; TEST SEQUENCE
          ;3188 ;  1) WRITE TO I/O SPACE TO CLEAR IMPLICT
          ;3189 ;       INTERLOCK FLIP/FLOP
          ;3190 ;  2) READ.V.WCHK TO I/O SPACE TO SET
          ;3191 ;       IMPLICT INTERLOCK FLOP
          ;3192 ;  3) ENABLE SILO, WRITE.P TO I/O SPACE
          ;3193 ;  4) READ SILO SEE INTERLOCK WRITE COMMAND-
          ;3194 ;       ADDRESS (C/A) IN SILO
          ;3195 ;  5) ENABLE SILO, WRITE.P TO I/O SPACE
          ;3196 ;  6) READ SILO, SEE MASKED WRITE OPERATION
          ;3197 ;       (C/A) IN SILO
          ;3198 ;
```

```
            ;3199 ;
            ;3200 ; ERROR LOGOUT
            ;3201 ;
            ;3202 ;  UNDEFINED
            ;3203 ;  UNDEFINED
            ;3204 ;  UNDEFINED
            ;3205 ;  UNDEFINED
            ;3206 ;  UNDEFINED
            ;3207 ;  UNDEFINED
            ;3208 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
            ;3209 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
            ;3210 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
            ;3211 ;  INTERRUPT FLAG:
            ;3212 ;   0=>NO INTERRUPT OCCURRED
            ;3213 ;   1=>AN INTERRUPT OCCURRED
            ;3214 ;
            ;3215 ;
            ;3216 ;-
            ;3217 ;================================================================================
            ;3218 ;
U 10C4, 0000.003D.0180.0800.0000.11C0  ;3219 =0 SUBTEST
U 10C5, 0000.003C.0180.0800.0000.10C6  ;3220  NOP
U 10C6, 0000.003D.0180.0800.0000.11BA  ;3221 =0 ERLOOP
U 10C7, 0000.003C.0180.0800.0000.10C8  ;3222  NOP
U 10C8, 0000.003D.F580.0800.0084.71C4  ;3223 =0 SETRCF[.A]
U 10C9, 0000.003C.0180.0800.0000.10CA  ;3224  NOP
U 10CA, 0000.003D.0180.0800.0000.124D  ;3225 =0 CALL[INIT]
U 10CB, 0F10.0038.0180.0970.0200.12B7  ;3226  VA_RC[0E].D_0   ;I/O WRITE TO
U 12B7, 0000.003C.0180.A800.0000.10CC  ;3227  MCT/WRITE.P   ;CLEAR IMPLICT
            ;3228 =0 D_K[.6],   ;INTERLOCK FLIP/FLOP
U 10CC, 0818.0039.D580.0800.0000.124B  ;3229   CALL[WAIT.LOOP]
U 10CD, 0000.003C.0180.0800.0000.1028  ;3230  NOP
U 1028, 0000.003D.0180.0800.0000.108C  ;3231 =00 CALL[NOINTR]
U 1029, 0000.003D.0180.0800.0000.11BB  ;3232  ERROR1    ;UNEXPECTD INTERRUPT
U 102A, 0010.0038.0180.0970.0200.12B8  ;3233  VA_RC[0E]
U 12B8, 0000.003C.0180.5000.0000.102C  ;3234 = MCT/READ.V.WCHK   ;OPERATION TO SET
U 102C, 0000.003D.0180.0800.0000.108C  ;3235 =00 CALL[NOINTR]   ; IMPLICT F/F
U 102D, 0000.003D.0180.0800.0000.11BB  ;3236  ERROR1    ;UNEXPECTED INTERRUPT!!!!!
U 102E, 0800.003C.0180.0A58.0000.12B9  ;3237  D_R[0B]    ;ENABLE SILO
U 12B9, 0000.003C.7180.3C00.0000.10CE  ;3238 = ID[COMP]_D
U 10CE, 0F10.0039.F580.0970.0284.71C4  ;3239 =0 VA_RC[0E].D_0,SETRCF[.A]
U 10CF, 0000.003C.0180.A800.0000.10D0  ;3240  MCT/WRITE.P   ;THIS IS THE WRITE THAT
            ;3241       ; SHOULD BE TRANSFORMED
            ;3242       ; TO INTERLOCK WRITE
            ;3243 =0 D_K[.10],   ; Wait
U 10D0, 0818.0039.6580.0800.0000.124B  ;3244   CALL[WAIT.LOOP]  ; ...
U 10D1, 0000.003C.0180.0800.0000.1030  ;3245  NOP
U 1030, 0000.003D.0180.0800.0000.1257  ;3246 =00 CALL[SILINT]   ;CHECK FOR SILO INTERRUPT
U 1031, 0000.003D.0180.0800.0000.11BB  ;3247  ERROR1    ;NOT SILO INTERRUPT
            ;3248   ;=============================================================
            ;3249   ;
            ;3250   ;READ SILO AND CHECK FOR INTERLOCK
            ;3251   ;WRITE COMMAND. IF NO COMPARE AFTER
            ;3252   ;READING ALL OF SILO MAKE ERROR CALL.
            ;3253   ;I.E. PREVIOUS WRITE OPERATION NO TRANSFORMED
```

```
            ;3254   ; TO INTERLOCK OPERATION
            ;3255   ;
            ;3256   ;================================================================
U 1032, 0003.003C.0180.09D0.0000.1033  ;3257  RC[0A]_0
U 1033, 0000.003C.61F0.2C00.0000.12BA  ;3258 SBX010: Q_ID[SILO]
U 12BA, 0800.003C.0180.0A00.0000.12BB  ;3259  D_R[0]    ;SET  INTERLOCK WRITE COMMAND
U 12BB, 0819.0030.0580.0800.0000.12BC  ;3260  D_D[OR]K[.1]   ;SET HOLD
U 12BC, 001D.2020.0180.0800.0010.12BD  ;3261  ALU_Q[XOR]D,CLK.UBCC  ;INTERLOCK WRITE?
U 12BD, 0000.013C.0180.0800.0000.10D2  ;3262  Z?
U 10D2, 0000.003C.0180.0800.0000.12BE  ;3263 =0 J/SBX020   ;NO CHECK COUNT
U 10D3, 0000.003C.0180.0800.0000.12C1  ;3264  J/SBX030   ;YES, INTERLOCK WRITE
U 12BE, 0810.0038.0180.0950.0000.12BF  ;3265 SBX020: D_RC[0A]
U 12BF, 0019.0020.6180.0800.0010.12C0  ;3266  ALU_D[XOR]K[.F],CLK.UBCC
U 12C0, 0819.0114.0580.09D0.0000.10D4  ;3267  Z?,RC[0A]_ALU,D_D+K[.1]
U 10D4, 0000.003C.0180.0800.0000.1033  ;3268 =0 J/SBX010
U 10D5, 0000.003D.0180.0800.0000.11BB  ;3269  ERROR1    ;NO INTERLOCK WRITE IN SILO
            ;3270       ; WRITE OPERATION WAS NOT
            ;3271       ; TRANSFORMED TO AND INTERLOCK
            ;3272       ; WRITE WHEN PRECEED BY A
            ;3273       ; READ.V.WCHK TO I/O SPACE
            ;3274   ;================================================================
            ;3275   ;
            ;3276   ;CHECK IMPLICT TO SEE THAT IT
            ;3277   ;     CLEARED AFTER THE TRANSFORMED WRITE
            ;3278   ;     OPERATION
            ;3279   ;
            ;3280   ;================================================================
U 12C1, 0800.003C.0180.0A58.0000.12C2  ;3281 SBX030: D_R[0B]    ;ENABLE SILO
U 12C2, 0000.003C.7180.3C00.0000.12C3  ;3282  ID[COMP]_D
U 12C3, 0F10.0038.0180.0970.0200.12C4  ;3283  VA_RC[0E],D_0
U 12C4, 0000.003C.0180.A800.0000.10D6  ;3284  MCT/WRITE.P
            ;3285 =0 D_K[.19],   ;
U 10D6, 0818.0039.B980.0800.0000.124B  ;3286   CALL[WAIT.LOOP]  ; LET SILO FILL
U 10D7, 0000.003C.0180.0800.0000.1034  ;3287  NOP
U 1034, 0000.003D.0180.0800.0000.1257  ;3288 =00 CALL[SILINT]
U 1035, 0000.003D.0180.0800.0000.11BB  ;3289  ERROR1    ;NOT A SILO INTERRUPT
            ;3290   ;================================================================
            ;3291   ;READ SILO, SEE MASKED WRITE C/A
            ;3292   ;================================================================
U 1036, 0003.003C.0180.09D0.0000.1037  ;3293  RC[0A]_0
U 1037, 0000.003C.61F0.2C00.0000.12C5  ;3294 SBX040: Q_ID[SILO]   ;READ SILO
U 12C5, 0800.003C.0180.0A20.0000.12C6  ;3295  D_R[4]    ;SET MASKED WRITE
U 12C6, 0819.0030.0580.0800.0000.12C7  ;3296  D_D[OR]K[.1]   ;SET HOLD
U 12C7, 001D.2020.0180.0800.0010.12C8  ;3297  ALU_Q[XOR]D,CLK.UBCC
U 12C8, 0000.013C.0180.0800.0000.10D8  ;3298  Z?
U 10D8, 0000.003C.0180.0800.0000.12C9  ;3299 =0 J/SBX050   ;NOT A MASKED WRITE
U 10D9, 0000.003C.0180.0800.0000.12CC  ;3300  J/SBX060   ;GOT THE MASKED WRITE
U 12C9, 0810.0038.0180.0950.0000.12CA  ;3301 SBX050: D_RC[0A]
U 12CA, 0019.0020.6180.0800.0010.12CB  ;3302  ALU_D[XOR]K[.F],CLK.UBCC
U 12CB, 0819.0114.0580.09D0.0000.10DA  ;3303  Z?,RC[0A]_ALU,D_D+K[.1]
U 10DA, 0000.003C.0180.0800.0000.1037  ;3304 =0 J/SBX040   ;REPEAT LOOP
U 10DB, 0000.003D.0180.0800.0000.11BB  ;3305  ERROR1    ;NO MASKED WRITE IN THE SILO !
            ;3306       ; 2ND WRITE AFTER IMPLICT INTERLOCK
            ;3307       ; OPERATION WAS NOT A MASKED
            ;3308       ; WRITE OPERATION. THE IMPLICTED
```

```
                ;3309       ; INTERLOCK FLOP WAS MOST LIKELY
                ;3310       ; NOT CLEARED AFTER THE 1ST
                ;3311       ; WRITE OPERATION.
U 12CC, 0000,003C,0180,0800,0000,10DC  ;3312 SBX060: NOP     ;END OF TEST
                ;3313 ;
                ;3314 ;
                ;3315 ;=================================================================================
                ;3316 ;*
                ;3317 ; SUBTEST 3 INHIBITING OF IMPLICT INTERLOCK
                ;3318 ;
                ;3319 ;  THIS TEST CHECKS THAT AN READ.V.WCHK DOES
                ;3320 ;  NOT SET THE IMPLICT INTERLOCK FLIP/FLOP
                ;3321 ;  WHEN A TIMEOUT OCCURES.  ANY ERROR THAT PREVENTS
                ;3322 ;  GOOD DATA FROM BEING RETURNED WILL PREVENT
                ;3323 ;  THE IMPLICT FLIP/FLOP FROM SETTING.  ONLY THE
                ;3324 ;  TIME OUT ERROR WILL BE CHECKED.
                ;3325 ;
                ;3326 ; TEST SEQUENCE
                ;3327 ;  1) WRITE TO I/O SPACE (CLEAR IMPLICT FLOP)
                ;3328 ;  2) SET ADDRESS TO CONFIG REG C + 4, CATCH TIMEOUTS
                ;3329 ;     READ.V.WCHK AND CHECK FOR TIMEOUT.
                ;3330 ;  3) ENABLE SILO, WRITE TO I/O SPACE
                ;3331 ;  4) READ SILO SEE A MASKED WRITE (NOT
                ;3332 ;     INTERLOCKED WRITE (C/A IN SILO)
                ;3333 ;
                ;3334 ;
                ;3335 ;ERROR LOGOUT
                ;3336 ;
                ;3337 ;  UNDEFINED
                ;3338 ;  UNDEFINED
                ;3339 ;  UNDEFINED
                ;3340 ;  UNDEFINED
                ;3341 ;  UNDEFINED
                ;3342 ;  UNDEFINED
                ;3343 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
                ;3344 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
                ;3345 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT ELSE UNDEFINED
                ;3346 ;  INTERRUPT FLAG:
                ;3347 ;   0=>NO INTERRUPT OCCURRED
                ;3348 ;   1=>AN INTERRUPT OCCURRED
                ;3349 ;
                ;3350 ;-
                ;3351 ;=================================================================================
                ;3352 ;
U 10DC, 0000,003D,0180,0800,0000,11C0  ;3353 =0 SUBTEST
U 10DD, 0000,003C,0180,0800,0000,10DE  ;3354  NOP
U 10DE, 0000,003D,0180,0800,0000,11BA  ;3355 =0 ERLOOP
U 10DF, 0000,003C,0180,0800,0000,10E0  ;3356  NOP
U 10E0, 0000,003D,0180,0800,0000,124D  ;3357 =0 CALL[INIT]
U 10E1, 0F10,0038,0180,0970,0200,12CD  ;3358  VA_RC[0E].D_0   ;CLEAR IMPLICT WITH
U 12CD, 0000,003C,0180,A800,0000,10E2  ;3359  MCT/WRITE.P   ;I/O WRITE OPERATION
                ;3360 =0 D_K[.6],
U 10E2, 0818,0039,D580,0800,0000,124B  ;3361   CALL[WAIT.LOOP]
U 10E3, 0000,003C,0180,0800,0000,1038  ;3362  NOP
U 1038, 0000,003D,0180,0800,0000,108C  ;3363 =00 CALL[NOINTR]
```

```
U 1039. 0000.003D.0180.0800.0000.11BB  ;3364  ERROR1     ;UNEXPECTED INTERRUPT
U 103A. 0000.003C.0180.0800.0000.103B  ;3365  NOP
U 103B. 0810.0038.0180.0970.0000.12CE  ;3366  D_RC[0E]     ;SET ADDRESS FOR TIMEOUT
U 12CE. 0819.0014.7580.0800.0200.10E4  ;3367  VA_ALU,D_D+K[.20]
U 10E4. 0000.003D.F580.0800.0084.71C4  ;3368 =0 SETRCF[.A]
U 10E5. 0F00.003C.8980.0800.0084.72CF  ;3369  SC_K[.D],D_0    ;CATCH TIMEOUT
U 12CF. 0801.001C.0180.0AF8.0000.12D0  ;3370  R[0F]_ALU,D_D.ORNOT.MASK
U 12D0. 0F00.003C.0180.0800.0000.12D1  ;3371  D_0
U 12D1. 0000.003C.0180.5000.0000.12D2  ;3372  MCT/READ.V.WCHK    ;READ, TRY TO SET IMPLICT
U 12D2. 0000.003C.0180.0A78.0010.12D3  ;3373  ALU_R[0F],CLK.UBCC
U 12D3. 0000.013C.0180.0800.0000.10E6  ;3374  Z?
U 10E6. 0000.003D.0180.0800.0000.11BB  ;3375 =0 ERROR1     ;DIDN'T GET TIMEOUT
U 10E7. 0000.003C.0180.0800.0000.10E8  ;3376  NOP
U 10E8. 0000.003D.0180.0800.0000.1260  ;3377 =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BITS
U 10E9. 0000.003C.0180.0800.0000.10EA  ;3378  NOP
U 10EA. 0000.003D.0180.0800.0000.1221  ;3379 =0 CALL[ENBFLT]    ;ENABLE CPU FAUT
U 10EB. 0800.003C.0180.0A58.0000.12D4  ;3380  D_R[0B]     ;     INTERRUPT
U 12D4. 0F10.0038.7180.3D70.0200.12D5  ;3381  ID[COMP]_D,VA_RC[0E],D_0  ;ENABLE SILO
U 12D5. 0000.003C.0180.A800.0000.10EC  ;3382  MCT/WRITE.P
        ;3383 =0 D_K[.19],    ;
U 10EC. 0818.0039.B980.0800.0000.124B  ;3384   CALL[WAIT.LOOP]  ; LET SILO FILL
U 10ED. 0000.003C.0180.0800.0000.103C  ;3385  NOP
U 103C. 0000.003D.0180.0800.0000.1257  ;3386 =00 CALL[SILINT]
U 103D. 0000.003D.0180.0800.0000.11BB  ;3387  ERROR1     ;GOT WRONG INTERRUPT
        ;3388    ;=============================================================
        ;3389    ;
        ;3390    ;READ CONTENTS OF SILO, SEE A MASKED WRITE C/A
        ;3391    ;     INSTEAD OF INTERLOCK WRITE
        ;3392    ;
        ;3393    ;=============================================================
U 103E. 0003.003C.0180.09D0.0000.103F  ;3394  RC[0A]_0
U 103F. 0000.003C.61F0.2C00.0000.12D6  ;3395 SBB010: Q_ID[SILO]
U 12D6. 0800.003C.0180.0A20.0000.12D7  ;3396  D_R[4]     ;GET MASKED WRITE COMMAND
U 12D7. 0819.0030.0580.0800.0000.12D8  ;3397  D_D[OR]K[.1]    ; AND SET HOLD BIT
U 12D8. 001D.0020.0180.0800.0010.12D9  ;3398  ALU_D[XOR]Q,CLK.UBCC   ;SEE MASKED WRITE C/A AND
U 12D9. 0810.0138.0180.0950.0000.10EE  ;3399  Z?,D_RC[0A]    ; HOLD
U 10EE. 0000.003C.0180.0800.0000.12DA  ;3400 =0 J/SBB020    ;NO
U 10EF. 0000.003C.0180.0800.0000.12DC  ;3401  J/SBB030    ;YES, MASKED WRITE C/A
U 12DA. 0019.0020.6180.0800.0010.12DB  ;3402 SBB020: ALU_D[XOR]K[.F],CLK.UBCC
U 12DB. 0819.0114.0580.09D0.0000.10F0  ;3403  Z?,RC[0A]_ALU,D_D+K[.1]
U 10F0. 0000.003C.0180.0800.0000.103F  ;3404 =0 J/SBB010    ;REPEAT LOOP
U 10F1. 0000.003D.0180.0800.0000.11BB  ;3405  ERROR1     ;NO MASKED WRITE IN SILO !
        ;3406       ; DIDN'T READ A MASKED
        ;3407       ; WRITE OPERATION (C/A)
        ;3408       ; FROM SILO, MOST LIKELY
        ;3409       ; THE IMPLICT GOT SET
        ;3410       ; AND INTERLOCK OPERATION
        ;3411       ; RESULTED.
U 12DC. 0000.003C.0180.0800.0000.10F2  ;3412 SBB030: NOP     ;END TEST
        ;3413 ;
        ;3414 ;=========================================================================
        ;3415 ;*
        ;3416 ; SUBTEST 4 RETRY TEST USING THE SILO
        ;3417 ;
        ;3418 ;  THIS TEST CHECKS THAT THE CPU SBL INTERFACE
```

```
;3419 ;  GENERATES THE CORRECT RETRY SEQUENCE OF EVENTS.
;3420 ;  THE SILO WILL BE ENABLE AND CONTENTS CLEARED
;3421 ;  AFTER THE SILO LOCKS. THE OPERATION USED TO
;3422 ;  CAUSE THE RETRY WILL BE A READ OPERATION TO
;3423 ;  A NON-EXISTANT I/O LOCATION (RC 2 + 4). THE
;3424 ;  SILO WILL BE CHECKED FOR 3(<CPU ARB>-C/A-NULL-
;3425 ;  ACK) SEQUENCES:
;3426 ;
;3427 ;
;3428 ;  SILO CONTENTS AFTER TIMEOUT.
;3429 ;
;3430 ;  NULL[ 2 LOCATIONS]
;3431 ;  CPU ARB (IF ID NOT=10)  --!
;3432 ;  C/A      !REPEATED 3 TIMES
;3433 ;  NULL      !
;3434 ;  NO-RESPONSE (NULL) CONFIRMATION   !
;3435 ;  NULL(WHILE CPU ANALYZES THE N.R.)-!
;3436 ;
;3437 ;
;3438 ;ERROR LOGOUT
;3439 ;
;3440 ;  DATA:  EXPECTED DATA
;3441 ;         DATA FROM SILO
;3442 ;         SILO COUNT, 0=1ST LOCATION AFTER LOCK(0-F)
;3443 ;         RETRY COUNT (0-2)
;3444 ;   UNDEFINED
;3445 ;   UNDEFINED
;3446 ;   ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3447 ;   ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3448 ;   ID[FAULT] IFF UNEXPECTED INTERRUPT
;3449 ;   INTERRUPT FLAG:
;3450 ;    0=>NO INTERRUPT OCCURRED
;3451 ;    1=>AN INTERRUPT OCCURRED
;3452 ;
;3453 ;  IF NONE: SEE LISTING FOR FAILING EVENT
;3454 ;
;3455 ;-
;3456 ;================================================================================
;3457 ;
U 10F2, 0000,003D,0180,0800,0000,11C0  ;3458 =0 SUBTEST
U 10F3, 0000,003C,0180,0800,0000,10F4  ;3459  NOP
U 10F4, 0000,003D,0180,0800,0000,11BA  ;3460 =0 ERLOOP
U 10F5, 0000,003C,0180,0800,0000,10F6  ;3461  NOP
U 10F6, 0000,003D,0180,0800,0000,124D  ;3462 =0 CALL[INIT]
U 10F7, 0000,003C,0180,0800,0000,10F8  ;3463  NOP
U 10F8, 0000,003D,1980,0800,0084,71C4  ;3464 =0 SETRCF[ZERO]
U 10F9, 0810,0038,0180,0970,0000,12DD  ;3465  D_RC[0E]    ;SET ADDRESS FOR TIMEOUT
U 12DD, 0819,0014,7580,0800,0200,12DE  ;3466  VA_ALU,D_D+K[.20]
U 12DE, 0F00,003C,8980,0800,0084,72DF  ;3467  D_0,SC_K[.D]   ;CATCH TIMEOUT
U 12DF, 0801,001C,0180,0AF8,0000,12E0  ;3468  R[0F]_ALU,D_D.ORNOT.MASK
U 12E0, 0800,003C,0180,0A58,0000,12E1  ;3469  D_R[0B]    ;ENABLE SILO
U 12E1, 0000,003C,7180,3C00,0000,12E2  ;3470  ID[COMP]_D
U 12E2, 0000,003C,0180,C800,0000,12E3  ;3471  MCT/READ.P   ;READ TO FILL SILO
U 12E3, 0000,003C,0180,0A78,0010,12E4  ;3472  ALU_R[0F],CLK.UBCC
U 12E4, 0000,013C,0180,0800,0000,10FA  ;3473  Z?
```

```
U 10FA, 0000,003D,0180,0800,0000,11BB  ;3474 =0 ERROR1     ;NO TIMEOUT!
        ;3475   ;=============================================================
        ;3476   ; CHECK CONTENT OF SILO
        ;3477   ;=============================================================
U 10FB, 0003,003C,0180,09E0,0000,10FC  ;3478  RC[0C]_0    ;SET EXPECTED DATE
U 10FC, 0003,003D,F580,09D0,0084,71C4  ;3479 =0 RC[0A]_0,SETRCF[.A]  ;SET SILO COUNTER
U 10FD, 0003,003C,0180,09C8,0000,12E5  ;3480  RC[9]_0    ;SET RETRY COUNTER
        ;3481   ;=============================================================
        ;3482   ;SEE NULL CYCLES
        ;3483   ;=============================================================
U 12E5, 0000,003C,61F0,2C00,0000,12E6  ;3484 SBC010: Q_ID[SILO]
U 12E6, 0001,203C,0180,09D8,0010,12E7  ;3485  RC[0B]_Q,CLK.UBCC
U 12E7, 0000,013C,0180,0800,0000,10FE  ;3486  Z?
U 10FE, 0000,003D,0180,0800,0000,11BB  ;3487 =0 ERROR1     ;NOT A NULL CYCLE IN SILO
U 10FF, 0810,0038,0180,0950,0000,12E8  ;3488  D_RC[0A]
U 12E8, 0019,0020,0980,0800,0010,12E9  ;3489  ALU_D[XOR]K[.2],CLK.UBCC
U 12E9, 0000,013C,0180,0800,0000,110A  ;3490  Z?
U 110A, 0819,0014,0580,09D0,0000,12E5  ;3491 =0 J/SBC010,RC[0A]_ALU,D_D+K[.1]
        ;3492   ;=============================================================
        ;3493   ;SEE ARB(I.D. NOT=10)
        ;3494   ;=============================================================
U 110B, 0010,0038,01C0,0910,0000,12EA  ;3495 SBC020: Q_RC[2]
U 12EA, 0019,2020,6580,0800,0010,12EB  ;3496  ALU_Q.XOR.K[.10],CLK.UBCC
U 12EB, 0000,013C,0180,0800,0000,1110  ;3497  Z?
U 1110, 0000,003C,0180,0800,0000,12EC  ;3498 =0 J/SBC030    ;I.D.NOT= 10
U 1111, 0000,003C,0180,0800,0000,1113  ;3499  J/SBC040    ;I.D.= 10
U 12EC, 0819,0014,0580,09D0,0000,12ED  ;3500 SBC030: RC[0A]_ALU,D_D+K[.1]
U 12ED, 0800,003C,61F0,2E50,0000,12EE  ;3501  D_R[0A],Q_ID[SILO]
U 12EE, 0001,003C,0180,09E0,0000,12EF  ;3502  RC[0C]_D
U 12EF, 0001,203C,0180,09D8,0000,12F0  ;3503  RC[0B]_Q
U 12F0, 001D,0020,0180,0800,0010,12F1  ;3504  ALU_D[XOR]Q,CLK.UBCC
U 12F1, 0810,0138,0180,0950,0000,1112  ;3505  Z?,D_RC[0A]
U 1112, 0000,003D,0180,0800,0000,11BB  ;3506 =0 ERROR1     ;DIDN'T SEE CPU ARB IN SILO
        ;3507   ;=============================================================
        ;3508   ;SEE MASKED READ C/A
        ;3509   ;=============================================================
U 1113, 0819,0014,0580,09D0,0000,12F2  ;3510 SBC040: RC[0A]_ALU,D_D+K[.1]
U 12F2, 0800,003C,61F0,2E18,0000,12F3  ;3511  D_R[3],Q_ID[SILO]
U 12F3, 0001,003C,0180,09E0,0000,12F4  ;3512  RC[0C]_D
U 12F4, 0001,203C,0180,09D8,0000,12F5  ;3513  RC[0B]_Q
U 12F5, 001D,0020,0180,0800,0010,12F6  ;3514  ALU_D[XOR]Q,CLK.UBCC
U 12F6, 0810,0138,0180,0950,0000,1114  ;3515  Z?,D_RC[0A]
U 1114, 0000,003D,0180,0800,0000,11BB  ;3516 =0 ERROR1     ;DIDN'T SEE MASKED READ IN SILO
        ;3517   ;=============================================================
        ;3518   ;SEE NULL CYCLE
        ;3519   ;=============================================================
U 1115, 0819,0014,0580,09D0,0000,12F7  ;3520  RC[0A]_ALU,D_D+K[.1]
U 12F7, 0003,003C,61F0,2DE0,0000,12F8  ;3521  RC[0C]_0,Q_ID[SILO]
U 12F8, 0001,203C,0180,09D8,0010,12F9  ;3522  RC[0B]_Q,CLK.UBCC
U 12F9, 0810,0138,0180,0950,0000,1116  ;3523  Z?,D_RC[0A]
U 1116, 0000,003D,0180,0800,0000,11BB  ;3524 =0 ERROR1     ;DIDN'T SEE NULL CYCLE IN SILO
        ;3525   ;=============================================================
        ;3526   ;SEE NO RESPONSE CONFIRMATION
        ;3527   ;=============================================================
U 1117, 0819,0014,0580,09D0,0000,12FA  ;3528  RC[0A]_ALU,D_D+K[.1]
```

```
U 12FA. 0003,003C,61F0,2DE0,0000,12FB  ;3529  RC[0C]_0,Q_ID[SILO]
U 12FB. 0001,203C,0180,09D8,0010,12FC  ;3530  RC[0B]_Q,CLK.UBCC
U 12FC. 0810,0138,0180,0950,0000,1118  ;3531  Z?,D_RC[0A]
U 1118. 0000,003D,0180,0800,0000,11BB  ;3532 =0 ERROR1    ;DIDN'T SEE NO RESPONSE
          ;3533       ; CONFIRMATION IN SILO
          ;3534    ;================================================================
          ;3535    ;SEE NULL WHILE CPU DECODES THE NO-RESPONSE
          ;3536    ;================================================================
U 1119. 0819,0014,0580,09D0,0000,12FD  ;3537  RC[0A]_ALU,D_D+K[.1]  ;UPDATE COUNTER
U 12FD. 0003,003C,0180,09E0,0000,12FE  ;3538  RC[0C]_0
U 12FE. 0000,003C,61F0,2C00,0000,12FF  ;3539  Q_ID[SILO]
U 12FF. 0001,203C,0180,09D8,0010,1300  ;3540  RC[0B]_Q,CLK.UBCC  ;CHECK FOR NULL
U 1300. 0810,0138,0180,0950,0000,111A  ;3541  Z?,D_RC[0A]
U 111A. 0000,003D,0180,0800,0000,11BB  ;3542 =0 ERROR1    ;NO NULL IN SILO
          ;3543    ;================================================================
          ;3544    ;CHECK RETRY COUNTER
          ;3545    ;================================================================
U 111B. 0010,0038,01C0,0948,0000,1301  ;3546  Q_RC[9]
U 1301. 0019,2020,0980,0800,0010,1302  ;3547  ALU_Q.XOR.K[.2],CLK.UBCC
U 1302. 0019,2114,05C0,0800,0000,111C  ;3548  Z?,Q_Q+K[.1]
U 111C. 0001,203C,0180,09C8,0000,110B  ;3549 =0 J/SBC020,RC[9]_Q  ;REPEAT LOOP
U 111D. 0000,003C,0180,0800,0000,111E  ;3550  NOP    ;END TEST
          ;3551 ;
          ;3552 ;=============================================================================
          ;3553 ;*
          ;3554 ; SUBTEST 5 UNEXPECTED READ DATA FAULT
          ;3555 ;
          ;3556 ;  UNEXPECTED READ DATA FAULT TO
          ;3557 ;  CPU. CHECK TO SEE THAT THE CPU SBI
          ;3558 ;  INTERFACE LOGIC CAN DETECT UNEXPECTED
          ;3559 ;  READ DATA FAULTS.  THE CPU ID (RC 3)
          ;3560 ;  WILL BE SET AS THE MAINTENANCE ID AND
          ;3561 ;  THE UNEXPECTED READ DATA BIT (ID=1D,#29) WILL
          ;3562 ;  BE SET.  THE SBI FAULT INTERRUPT WILL BE
          ;3563 ;  CHECKED AS WELL AS THE UNEXPECTED READ DATA
          ;3564 ;  BIT IN THE FAULT (BIT #28 ID=13) REGISTER.
          ;3565 ;  THE CACHE WILL BE DISABLED.
          ;3566 ;
          ;3567 ; ERROR LOGOUT
          ;3568 ;
          ;3569 ; IF NO PRINT OUT INDIVIDUAL BIT(CPU,SBI FAULT BIT)IN ERROR
          ;3570 ;
          ;3571 ; DATA: EXPECTED DATA
          ;3572 ;  MASKED IMAGE OF SBI-FAULT-STATUS REGISTER
          ;3573 ;  UNDEFINED
          ;3574 ;  UNDEFINED
          ;3575 ;  UNDEFINED
          ;3576 ;  UNDEFINED
          ;3577 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
          ;3578 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
          ;3579 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
          ;3580 ;  INTERRUPT FLAG:
          ;3581 ;   0=>NO INTERRUPT OCCURRED
          ;3582 ;   1=>AN INTERRUPT OCCURRED
          ;3583 ;
```

```
            ;3584 ;-
            ;3585 ;=================================================================================
            ;3586 ;
            ;3587 ;
U 111E, 0000,003D,0180,0800,0000,11C0  ;3588 =0 SUBTEST
U 111F, 0000,003C,0180,0800,0000,1120  ;3589  NOP
U 1120, 0000,003D,0180,0800,0000,11BA  ;3590 =0 ERLOOP
U 1121, 0000,003C,0180,0800,0000,1122  ;3591  NOP
U 1122, 0000,003D,0180,0800,0000,124D  ;3592 =0 CALL[INIT]
U 1123, 0000,003C,7DF8,0800,0084,7303  ;3593  SC_K[.18],Q_0    ;SHIFT ID TO
U 1303, 0810,0038,0580,0910,0084,B304  ;3594  SC_SC-K[.1],D_RC[2]  ; MAINTENANCE ID
U 1304, 0D00,003C,0180,0800,0000,1305  ;3595  D_DAL.SC    ; POSITION
U 1305, 0000,003C,75F0,2C00,0000,1306  ;3596  Q_ID[MAINT]    ;GET CACHE OFF BITS
U 1306, 081D,0030,ED80,0800,0084,7307  ;3597  D_D[OR]Q,SC_K[.1B]  ;SET FORCE
U 1307, 0000,003C,0980,0800,0084,9308  ;3598  SC_SC+K[.2]    ;  UNEXPECTED READ DATA
U 1308, 0801,001C,0180,0800,0000,1309  ;3599  D_D.ORNOT.MASK
U 1309, 0000,003C,7580,3C00,0000,1124  ;3600  ID[MAINT]_D    ;FORCE FAULT
            ;3601 =0 D_K[.6],    ; Wait
U 1124, 0818,0039,D580,0800,0000,124B  ;3602   CALL[WAIT.LOOP]  ; ...
U 1125, 0000,003C,0180,0800,0000,1126  ;3603  NOP
U 1126, 0000,003D,0180,0800,0000,11B6  ;3604 =0 CALL[DISABLE.CACHE]  ; Disable Cache and clear "Force
            ;3605       ; ...Fault" bits
U 1127, 0000,003C,0180,0800,0000,1040  ;3606  NOP
U 1040, 0000,003D,0180,0800,0000,1250  ;3607 =00 CALL[SBIFLT]    ;SEE SBI FAULT
U 1041, 0000,003D,F580,0800,0084,7064  ;3608  ERROR1[.A]    ;DIDN'T GET SBI FAULT INTERRUPT !!
U 1042, 0000,003C,6DF0,2C00,0000,1043  ;3609  Q_ID[FAULT]    ; Get the FAULT register
U 1043, 0001,203C,0180,09D8,0000,130A  ;3610  RC[0B]_Q    ; Save in RC[0B]
U 130A, 0818,0038,4580,0800,0000,130B  ;3611  D_K[.8000]
U 130B, 0110,0038,01C0,0958,0000,130C  ;3612  D_D.LEFT2,Q_RC[0B]  ;SEE SBI FAULT SET
U 130C, 001D,0034,0180,0800,0010,130D  ;3613  ALU_D[AND]Q,CLK.UBCC
U 130D, 0000,013C,0180,0800,0000,1128  ;3614  Z?
U 1128, 0000,003C,0180,0800,0000,130E  ;3615 =0 J/M19E00
U 1129, 0000,003D,1980,0800,0084,7064  ;3616  ERROR1[ZERO]    ;SBI FAULT NOT SET
U 130E, 0100,003C,0180,0800,0000,130F  ;3617 M19E00: D_D.LEFT2    ;CHECK  CPU FAULT
U 130F, 001D,0034,0180,0800,0010,1310  ;3618  ALU_D[AND]Q,CLK.UBCC  ;  SET
U 1310, 0000,013C,0180,0800,0000,112A  ;3619  Z?
U 112A, 0000,003C,0180,0800,0000,1311  ;3620 =0 J/M19E01
U 112B, 0000,003D,1980,0800,0084,7064  ;3621  ERROR1[ZERO]    ;CPU FAULT NOT SET
U 1311, 0818,0038,CDF8,0800,0000,1312  ;3622 M19E01: D_K[.F0],Q_0
U 1312, 0000,003C,8580,0800,0084,7313  ;3623  SC_K[.C]
U 1313, 0D00,003C,0180,0800,0000,1314  ;3624  D_DAL.SC    ;MASK OFF BITS 16-19
U 1314, 0010,0038,01C0,0958,0000,1315  ;3625  Q_RC[0B]
U 1315, 001D,2024,01C0,09D8,0000,1316  ;3626  RC[0B]_ALU,Q_Q.ANDNOT.D
U 1316, 0000,003C,ED80,0800,0084,7317  ;3627  SC_K[.1B]
U 1317, 0F00,003C,0980,0800,0084,9318  ;3628  SC_SC+K[.2],D_0    ;
U 1318, 0801,001C,0180,09E0,0000,1319  ;3629  RC[0C]_ALU,D_D.ORNOT.MASK ;CHECK BIT #29
U 1319, 001D,0020,0180,0800,0010,131A  ;3630  ALU_D[XOR]Q,CLK.UBCC
U 131A, 0000,013C,0180,0800,0000,112C  ;3631  Z?
U 112C, 0000,003D,F580,0800,0084,7064  ;3632 =0 ERROR1[.A]    ;WRONG CPU-SBI-FAULT BITS SET
            ;3633       ;=====================================
            ;3634       ;DATA: EXPECTED DATA
            ;3635       ; MASK IMAGE CPU-FAULT REG
            ;3636       ;=====================================
U 112D, 0000,003C,0180,0800,0000,112E  ;3637  NOP
U 112E, 0000,003D,0180,0800,0000,1264  ;3638 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT
```

```
U 112F, 0818.0038.4580.0800.0000.131B  ;3639  D_K[.8000]
U 131B, 0100.003C.0180.0800.0000.131C  ;3640  D_D.LEFT2
U 131C, 0100.003C.6DF0.2C00.0000.131D  ;3641  Q_ID[FAULT],D_D.LEFT2
U 131D, 001D.0034.0180.0800.0010.131E  ;3642  ALU_D[AND]Q,CLK.UBCC  ;CPU FAULT CLEAR?
U 131E, 0000.013C.0180.0800.0000.1130  ;3643  Z?
U 1130, 0000.003D.1980.0800.0084.7064  ;3644 =0 ERROR1[ZERO]   ;CPU FAULT DID NOT CLEAR
U 1131, 0200.003C.0180.0800.0000.131F  ;3645  D_D.RIGHT2
U 131F, 001D.0034.0180.0800.0010.1320  ;3646  ALU_D[AND]Q,CLK.UBCC
U 1320, 0000.013C.0180.0800.0000.1132  ;3647  Z?    ;SBI FAULT CLEAR?
U 1132, 0000.003D.1980.0800.0084.7064  ;3648 =0 ERROR1[ZERO]   ;NO, SBI FAULT DID NOT CLEAR
U 1133, 0001.203C.01F8.0AF0.0000.1321  ;3649  R[0E]_Q,Q_0
U 1321, 0818.0038.CD80.0800.0000.1322  ;3650  D_K[.F0]
U 1322, 0000.003C.8580.0800.0084.7323  ;3651  SC_K[.C]
U 1323, 0D00.003C.0180.0800.0000.1324  ;3652  D_DAL.SC
U 1324, 0000.003C.01C0.0A70.0000.1325  ;3653  Q_R[0E]
U 1325, 081D.2024.0180.0800.0000.1326  ;3654  D_Q.ANDNOT.D   ;MASK OFF BITS 19-16
U 1326, 0001.003C.0180.09D8.0000.1327  ;3655  RC[0B]_D   ;SAVE MASKED IMAGE
U 1327, 001D.0020.0180.0800.0010.1328  ;3656  ALU_D[XOR]Q,CLK.UBCC  ;DID SBI-FAULT-STATUS CLEAR ?
U 1328, 0003.013C.0180.09E0.0000.1134  ;3657  Z?,RC[0C]_0
U 1134, 0000.003D.F580.0800.0084.7064  ;3658 =0 ERROR1[.A]   ;NO, CPU-SBI-FAULT-STATUS BITS
          ;3659       ;  DID NOT CLEAR
          ;3660       ;=====================================
          ;3661       ;DATA: EXPECTED DATA
          ;3662       ; MASKED IMAGE OF CPU-
          ;3663       ;      SBI-STATUS REGISTER
          ;3664       ;=====================================
U 1135, 0000.003C.0180.0800.0000.1136  ;3665  NOP
          ;3666 ;
          ;3667 ;
          ;3668 ; SUB002.MIC
          ;3669 ;
          ;3670 ;=========================================================================
          ;3671 ;*
          ;3672 ; SUBTEST 6
          ;3673 ;
          ;3674 ; CHECK SBI ERROR CONFIRMATION ON BYTE
          ;3675 ; WRITES TO I/O SPACE.  SEES THAT WRITES TO
          ;3676 ; I/O U-TRAP(STALL).
          ;3677 ;
          ;3678 ; ID=19, "SBI.ERR",BIT #8, CPU SBI ERROR CONFIRMATION
          ;3679 ;
          ;3680 ; ERROR LOGOUT
          ;3681 ;
          ;3682 ; DATA: BYTE DISPLACEMENT BEING USED OFF THE
          ;3683 ;       BASE OF THE  LONG WORD
          ;3684 ;  TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
          ;3685 ;  UNDEFINED
          ;3686 ;  UNDEFINED
          ;3687 ;  UNDEFINED
          ;3688 ;  UNDEFINED
          ;3689 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
          ;3690 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
          ;3691 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
          ;3692 ;  INTERRUPT FLAG:
          ;3693 ;   0=>NO INTERRUPT OCCURRED
```

```
            ;3694 ;   1=>AN INTERRUPT OCCURRED
            ;3695 ;
            ;3696 ;-
            ;3697 ;================================================================================
            ;3698 ;
U 1136. 0000.003D.0180.0800.0000.11C0  ;3699 =0 SUBTEST
U 1137. 0000.003C.0180.0800.0000.1138  ;3700  NOP
U 1138. 0000.003D.0180.0800.0000.11BA  ;3701 =0 ERLOOP
U 1139. 0000.003C.0180.0800.0000.113A  ;3702  NOP
U 113A. 0000.003D.0180.0800.0000.124D  ;3703 =0 CALL[INIT]
U 113B. 0000.003C.65F0.2C00.0000.1329  ;3704  Q_ID[SBI.ERR]   ;SEE ERR CONFIRMATION CLEAR
U 1329. 0000.003C.0180.0800.0084.732A  ;3705  SC_K[.8]
U 132A. 0001.2024.0180.0800.0010.113C  ;3706  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 113C. 0000.003D.F580.0800.0084.71C4  ;3707 =0 SETRCF[.A]
U 113D. 0000.013C.0180.0800.0000.113E  ;3708  Z?
U 113E. 0000.003D.0180.0800.0000.11BB  ;3709 =0 ERROR1    ;ERROR CONFIRMATION SET
            ;3710       ;  DIDN'T CLEAR ON INIT
U 113F. 0000.003C.91F0.2C00.0000.132B  ;3711  Q_ID[24]
U 132B. 0001.203C.0180.09D8.0000.1140  ;3712  RC[0B]_Q
U 1140. 0003.003D.F580.09E0.0084.71C4  ;3713 =0 RC[0C]_0,SETRCF[.A]  ;LOOP THROUGH 4 BYTE
U 1141. 0000.003C.0180.0800.0000.1142  ;3714  NOP    ;POSITIONS
U 1142. 0000.003D.0180.0800.0000.11BA  ;3715 =0 ERLOOP
U 1143. 0810.0038.0180.0960.0000.132C  ;3716 M19C01: D_RC[0C]
U 132C. 0010.0038.01C0.0970.0000.132D  ;3717  Q_RC[0E]
U 132D. 001D.0014.0180.0800.0200.132E  ;3718  VA_ALU,ALU_D[A+B]Q  ;FIND BYTE ADDRESS
U 132E. 0F00.003C.8980.0800.0084.732F  ;3719  D_0,SC_K[.D]   ;CATCH TIMEOUT
U 132F. 0801.001C.0180.0AF8.0000.1330  ;3720  R[0F]_ALU,D_D.ORNOT.MASK
U 1330. 0000.803C.0180.A800.0000.1331  ;3721  MCT/WRITE.P,BYTE
U 1331. 0000.003C.0180.0A78.0010.1332  ;3722  ALU_R[0F],CLK.UBCC  ;DID IT U-TRAP?
U 1332. 0003.013C.0180.0AF8.0000.1144  ;3723  Z?,R[0F]_0
U 1144. 0000.003D.0180.0800.0000.11BB  ;3724 =0 ERROR1    ;DIDN'T U-TRAP OR STALL
U 1145. 0000.003C.65F0.2C00.0000.1333  ;3725  Q_ID[SBI.ERR]   ;CHECK SBI ERROR CONFIRMATION
U 1333. 0000.003C.0180.0800.0084.7334  ;3726  SC_K[.8]   ; BIT SET
U 1334. 0001.2024.0180.0800.0010.1335  ;3727  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1335. 0000.013C.0180.0800.0000.1146  ;3728  Z?
U 1146. 0000.003C.0180.0800.0000.1148  ;3729 =0 J/M19C02   ;BIT WAS SET
U 1147. 0000.003D.0180.0800.0000.11BB  ;3730  ERROR1    ;CPU ERR CONFIRMATION
            ;3731       ; BIT WAS CLEAR
            ;3732 =0
U 1148. 0000.003D.0180.0800.0000.124D  ;3733 M19C02: CALL[INIT]   ;CLEAR CPU
U 1149. 0000.003C.65F0.2C00.0000.1336  ;3734  Q_ID[SBI.ERR]   ;SEE ERR CONFIRMATION
U 1336. 0000.003C.0180.0800.0084.7337  ;3735  SC_K[.8]   ; CLEAR
U 1337. 0001.2024.0180.0800.0010.1338  ;3736  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1338. 0000.013C.0180.0800.0000.114A  ;3737  Z?
U 114A. 0000.003D.0180.0800.0000.11BB  ;3738 =0 ERROR1    ;ERR CONFIRMATION DID
            ;3739       ; NOT CLEAR
U 114B. 0810.0038.0180.0960.0000.1339  ;3740  D_RC[0C]   ;CHECK FOR LAST BYTE
U 1339. 0019.0020.0D80.0800.0010.133A  ;3741  ALU_D.XOR.K[.3],CLK.UBCC
U 133A. 0000.013C.0180.0800.0000.114C  ;3742  Z?
U 114C. 0819.0014.0580.09E0.0000.1143  ;3743 =0 RC[0C]_ALU,D_D+K[.1],J/M19C01 ;INCREMENT BYTE DISPLACEMENT
U 114D. 0000.003C.0180.0800.0000.114E  ;3744  NOP    ;END OF TEST
            ;3745 ;
            ;3746 ;================================================================================
            ;3747 ;+
            ;3748 ;
```

```
            ;3749 ; SUBTEST 7
            ;3750 ;
            ;3751 ;  SEE THAT EXTENDED WRITES TO I/O SPACE
            ;3752 ;  STALL,U-TRAP AND SET SBI ERROR
            ;3753 ;  COMFIRMATION.
            ;3754 ;
            ;3755 ; ERROR LOGOUT
            ;3756 ;
            ;3757 ; NONE, SEE LISTING FOR FAILING EVENT
            ;3758 ;
            ;3759 ;-
            ;3760 ;=============================================================================
            ;3761 ;
U 114E. 0000.003D.0180.0800.0000.11C0  ;3762 =0 SUBTEST
U 114F. 0000.003C.0180.0800.0000.1150  ;3763  NOP
U 1150. 0000.003D.0180.0800.0000.11BA  ;3764 =0 ERLOOP
U 1151. 0000.003C.0180.0800.0000.1152  ;3765  NOP
U 1152. 0000.003D.0180.0800.0000.124D  ;3766 =0 CALL[INIT]
U 1153. 0010.0038.0180.0970.0200.133B  ;3767  VA_RC[0E]    ;SET I/O ADDRESS
U 133B. 0F00.003C.8980.0800.0084.7156  ;3768  D_0,SC_K[.D]    ;CATCH TIMOUT U-TRAPS
U 1156. 0801.001D.1980.0AF8.0084.71C4  ;3769 =0 R[0F]_ALU,D_D.ORNOT.MASK,SETRCF[ZERO]
U 1157. 0000.003C.0180.A000.0000.133C  ;3770  MCT/EXTWRITE.P
U 133C. 0000.003C.0180.0A78.0010.133D  ;3771  ALU_R[0F],CLK.UBCC
U 133D. 0000.013C.0180.0800.0000.1158  ;3772  Z?
U 1158. 0000.003D.0180.0800.0000.11BB  ;3773 =0 ERROR1     ;DID NOT U-TRAP
U 1159. 0000.003C.0180.0800.0084.733E  ;3774  SC_K[.8]    ;CHECK ERR CONF IN CPU
U 133E. 0000.003C.65F0.2C00.0000.133F  ;3775  Q_ID[SBI.ERR]    ;READ FAULT
U 133F. 0001.2024.0180.0800.0010.1340  ;3776  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1340. 0000.013C.0180.0800.0000.115A  ;3777  Z?
U 115A. 0000.003C.0180.0800.0000.1341  ;3778 =0 J/M19C03    ;BIT WAS SET
U 115B. 0000.003D.0180.0800.0000.11BB  ;3779  ERROR1     ;ERROR CONFIRMATION BIT DID NOT SET
U 1341. 0000.003C.0180.0800.0000.115C  ;3780 M19C03: NOP
            ;3781 ;  SUB003.MIC
            ;3782 ;
            ;3783 ;=============================================================
            ;3784 ;*
            ;3785 ; SUBTEST 8
            ;3786 ;
            ;3787 ;  CHECK CPU MULTIPLE ERROR BIT-ID REGISTER 19.
            ;3788 ;  "SBI.ERR",BIT #2.  SEE CLEAR AFTER INIT SEE
            ;3789 ;  SET BY TWO CPU TIMEOUTS CHECK AFTER 1ST ERROR THAT BIT
            ;3790 ;  IS NOT SET.  CHECK FOR SBI NOT-BUSY SET ALSO.
            ;3791 ;
            ;3792 ;-
            ;3793 ;=============================================================
            ;3794 ;
U 115C. 0000.003D.0180.0800.0000.11C0  ;3795 =0 SUBTEST
U 115D. 0000.003C.0180.0800.0000.115E  ;3796  NOP
U 115E. 0000.003D.0180.0800.0000.11BA  ;3797 =0 ERLOOP
U 115F. 0000.003C.0180.0800.0000.1160  ;3798  NOP
U 1160. 0000.003D.0180.0800.0000.124D  ;3799 =0 CALL[INIT]
U 1161. 0000.003C.0180.0800.0000.1162  ;3800  NOP
U 1162. 0000.003D.1980.0800.0084.71C4  ;3801 =0 SETRCF[ZERO]
U 1163. 0000.003C.65F0.2C00.0000.1342  ;3802  Q_ID[SBI.ERR]
U 1342. 0019.2034.1180.0800.0010.1343  ;3803  ALU_Q.AND.K[.4],CLK.UBCC
```

```
U 1343. 0000.013C.0180.0800.0000.1164  ;3804  Z?
U 1164. 0000.003D.0180.0800.0000.11BB  ;3805 =0 ERROR1     ;MULTIPLE ERROR BIT
        ;3806       ; DID NOT CLEAR
U 1165. 0019.2034.0980.0800.0010.1344  ;3807  ALU_Q.AND.K[.2],CLK.UBCC
U 1344. 0000.013C.0180.0800.0000.1166  ;3808  Z?
U 1166. 0000.003C.0180.0800.0000.1345  ;3809 =0 J/M19D00
U 1167. 0000.003D.0180.0800.0000.11BB  ;3810  ERROR1     ;SBI INTERFACE BUSY
        ;3811       ;  DID NOT CLEAR ON INIT
U 1345. 0000.003C.8580.0800.0084.7346  ;3812 M19D00: SC_K[.C]    ;CHECK FOR CPU TIMEOUT
U 1346. 0001.2024.0180.0800.0010.1347  ;3813  ALU_Q.ANDNOT.MASK,CLK.UBCC ;CHECK BIT #12
U 1347. 0000.013C.0180.0800.0000.1168  ;3814  Z?
U 1168. 0000.003D.0180.0800.0000.11BB  ;3815 =0 ERROR1     ;CPU TIMEOUT SET, NOT CLEAR
U 1169. 0810.0038.0180.0970.0000.1348  ;3816  D_RC[0E]     ; ON INIT
U 1348. 0819.0014.7580.0800.0200.1349  ;3817  VA_ALU,D_D+K[.20]  ;SET I/O ADDRESS FOR
        ;3818       ;   TIMEOUT
U 1349. 0F00.003C.8980.0800.0084.734A  ;3819  SC_K[.D],D_0    ;CATCH TIMEOUT
U 134A. 0801.001C.0180.0AF8.0000.134B  ;3820  R[0F]_ALU,D_D.ORNOT.MASK
U 134B. 0000.003C.0180.C800.0000.134C  ;3821  MCT/READ.P    ;READ CAUSING TIMEOUT(1ST)
U 134C. 0000.003C.65F0.2C00.0000.134D  ;3822  Q_ID[SBI.ERR]    ;READ ERROR REGISTER
U 134D. 0000.003C.8580.0800.0084.734E  ;3823  SC_K[.C]    ;
U 134E. 0001.2024.0180.0800.0010.134F  ;3824  ALU_Q.ANDNOT.MASK,CLK.UBCC ;CHECK FOR TIMEOUT
U 134F. 0000.013C.0180.0800.0000.116A  ;3825  Z?
U 116A. 0000.003C.0180.0800.0000.1350  ;3826 =0 J/M19D01    ;GOT TIMEOUT
U 116B. 0000.003D.0180.0800.0000.11BB  ;3827  ERROR1     ;CPU TIMEOUT BIT NOT SET
U 1350. 0000.003C.0180.0800.0084.7351  ;3828 M19D01: SC_K[.8]    ;CHECK MULTIPLE ERROR
U 1351. 0001.2024.0180.0800.0010.1352  ;3829  ALU_Q.ANDNOT.MASK,CLK.UBCC ;  BIT CLEAR
U 1352. 0000.013C.0180.0800.0000.116C  ;3830  Z?
U 116C. 0000.003D.0180.0800.0000.11BB  ;3831 =0 ERROR1     ;MULTIPLE ERROR BIT
        ;3832       ; SET ON 1ST ERROR
U 116D. 0F00.003C.8980.0800.0084.7353  ;3833  SC_K[.D],D_0    ;FORCE 2ND ERROR
U 1353. 0801.001C.0180.0AF8.0000.1354  ;3834  R[0F]_ALU,D_D.ORNOT.MASK ;CATCH TIMEOUT
        ;3835       ;VA STILL LOAD
U 1354. 0000.003C.0180.C800.0000.1355  ;3836  MCT/READ.P    ;  FOR TIMEOUT(2ND)
U 1355. 0000.003C.65F0.2C00.0000.1356  ;3837  Q_ID[SBI.ERR]
U 1356. 0019.2034.1180.0800.0010.1357  ;3838  ALU_Q.AND.K[.4],CLK.UBCC
U 1357. 0000.013C.0180.0800.0000.116E  ;3839  Z?
U 116E. 0000.003C.0180.0800.0000.1170  ;3840 =0 J/M19D03    ;GOOD, IT'S SET
U 116F. 0000.003D.0180.0800.0000.11BB  ;3841  ERROR1     ;MULTIPLE ERROR,NOT
        ;3842       ; SET ON 2ND TIMEOUT
        ;3843 =0
U 1170. 0000.003D.0180.0800.0000.124D  ;3844 M19D03: CALL[INIT]    ;INIT,SEE MULTIPLE
U 1171. 0000.003C.65F0.2C00.0000.1358  ;3845  Q_ID[SBI.ERR]    ; ERROR CLEAR
U 1358. 0019.2034.1180.0800.0010.1359  ;3846  ALU_Q.AND.K[.4],CLK.UBCC
U 1359. 0000.013C.0180.0800.0000.1172  ;3847  Z?
U 1172. 0000.003D.0180.0800.0000.11BB  ;3848 =0 ERROR1     ;MULTIPLE ERROR DID
        ;3849       ; NOT CLEAR
U 1173. 0000.003C.0180.0800.0000.1174  ;3850  NOP     ;END TEST
        ;3851 ;
        ;3852 ;=================================================================
        ;3853 ;
        ;3854 ; SUBTEST 9
        ;3855 ;
        ;3856 ; CHECK THE SETTING AND CLEARING OF THE
        ;3857 ; SBI-NOT-BUSY BIT IN THE ID=19,"SBI.ERR"(BIT #2)
        ;3858 ; REGISTER. THE BIT WILL BE CHECKED TO BE CLEAR
```

```
            ;3859 ; DURING A WRITE OPERATION.
            ;3860 ;
            ;3861 ; ERROR LOGOUT
            ;3862 ;
            ;3863 ; NONE
            ;3864 ;
            ;3865 ;-
            ;3866 ;=================================================================
            ;3867 ;
U 1174, 0000.003D.0180.0800.0000.11C0  ;3868 =0 SUBTEST
U 1175, 0000.003C.0180.0800.0000.1176  ;3869  NOP
U 1176, 0000.003D.0180.0800.0000.11BA  ;3870 =0 ERLOOP
U 1177, 0000.003C.0180.0800.0000.1178  ;3871  NOP
U 1178, 0000.003D.0180.0800.0000.124D  ;3872 =0 CALL[INIT]
U 1179, 0000.003C.0180.0800.0000.117A  ;3873  NOP
U 117A, 0000.003D.1980.0800.0084.71C4  ;3874 =0 SETRCF[ZERO]
U 117B, 0000.003C.65F0.2C00.0000.135A  ;3875  Q_ID[SBI.ERR]   ;READ SBI-NOT-BUSY BIT
U 135A, 0019.2034.0980.0800.0010.135B  ;3876  ALU_Q.AND.K[.2],CLK.UBCC ;IS IT SET ???
U 135B, 0000.013C.0180.0800.0000.117C  ;3877  Z?
U 117C, 0000.003C.0180.0800.0000.135C  ;3878 =0 J/M19D10    ;YES, IT'S SET
U 117D, 0000.003D.0180.0800.0000.11BB  ;3879  ERROR1     ;SBI-NOT-BUSY IS CLEAR
U 135C, 0018.0038.1980.0800.0200.135D  ;3880 M19D10: VA_K[ZERO]   ;DO A WRITE AND SEE IT CLEAR
U 135D, 0000.003C.0180.A800.0000.135E  ;3881  MCT/WRITE.P
U 135E, 0000.003C.0180.0800.0000.135F  ;3882  NOP
U 135F, 0000.003C.65F0.2C00.0000.1360  ;3883  Q_ID[SBI.ERR]
U 1360, 0019.2034.0980.0800.0010.1361  ;3884  ALU_Q.AND.K[.2],CLK.UBCC ;IS IT CLEAR ???
U 1361, 0000.013C.0180.0800.0000.117E  ;3885  Z?
U 117E, 0000.003D.0180.0800.0000.11BB  ;3886 =0 ERROR1     ;SBI-NOT-BUSY DID NOT CLEAR
            ;3887       ;  DURING A WRITE OPERATION
U 117F, 0000.003C.0180.0800.0000.1180  ;3888  NOP
            ;3889 ;
            ;3890 ;=================================================================
            ;3891 ;*
            ;3892 ; SUBTEST A STALL ON BACK TO BACK WRITES
            ;3893 ;
            ;3894 ; THIS SUBTEST CHECK THAT BACK TO BACK WRITES WORK PROPERLY.
            ;3895 ; BACK TO BACK WRITE WILL BE INITIATED WITH DIFFERENT
            ;3896 ; ADDRESS AND DATA. AFTER A 104 USEC WAIT, THE LOCATIONS WILL BE
            ;3897 ; READ TO SEE THE CORRECT DATA EXSISTS. ALSO INTERRUPTS WILL BE
            ;3898 ; CHECKED TO ASSURE THAT NO BUS FAULTS WERE DETECTED.
            ;3899 ;
            ;3900 ; ERROR LOGOUT
            ;3901 ;
            ;3902 ; DATA: EXPECTED DATA
            ;3903 ;  DATA READ
            ;3904 ;  UNDEFINED
            ;3905 ;  UNDEFINED
            ;3906 ;  UNDEFINED
            ;3907 ;  UNDEFINED
            ;3908 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;3909 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;3910 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;3911 ;  INTERRUPT FLAG:
            ;3912 ;   0=>NO INTERRUPT OCCURRED
            ;3913 ;   1=>AN INTERRUPT OCCURRED
```

```
            ;3914 ;
            ;3915 ;
            ;3916 ;-
            ;3917 ;================================================================================
            ;3918 ;
U 1180, 0000.003D.0180.0800.0000.11C0  ;3919 =0 SUBTEST
U 1181, 0000.003C.0180.0800.0000.1182  ;3920  NOP
U 1182, 0000.003D.0180.0800.0000.11BA  ;3921 =0 ERLOOP
U 1183, 0000.003C.0180.0800.0000.1184  ;3922  NOP
U 1184, 0000.003D.0180.0800.0000.124D  ;3923 =0 CALL[INIT]
U 1185, 0F18.0038.1980.0800.0200.1362  ;3924  VA_K[ZERO],D_0   ;SET BACKGROUD DATA FOR FIRST WRITE
U 1362, 0000.003C.0180.A800.0000.1186  ;3925  MCT/WRITE.P   ;SYNC:WRITE BACKGROUD DATA FOR 1ST WRITE
            ;3926 =0 D_K[.10],
U 1186, 0818.0039.6580.0800.0000.124B  ;3927   CALL[WAIT.LOOP]  ;
U 1187, 0F18.0038.7580.0800.0200.1363  ;3928  VA_K[.20],D_0   ;SET BACKGROUND DATA FOR 2ND WRITE
U 1363, 0000.003C.0180.A800.0000.1188  ;3929  MCT/WRITE.P   ;SYNC:WRITE BACKGROUD DATA FOR 2ND WRITE
U 1188, 0003.003D.F580.0800.0284.71C4  ;3930 =0 SETRCF[.A],ALU_0(A),VA_ALU ;1ST ADDRESS AND DATA FOR BACK TO
U 1189, 0818.0038.C180.0800.0000.1364  ;3931  D_K[.FFFF]   ;   BACK WRITES
U 1364, 0000.003C.0180.A800.0000.1365  ;3932  MCT/WRITE.P   ;SYNC:1ST WRITE(BACK TO BACK)
U 1365, 0801.0028.0180.0800.0000.1366  ;3933  D_NOT.D
U 1366, 0018.0038.7580.0800.0200.1367  ;3934  VA_K[.20]   ;ADDRESS AND DATA FOR 2ND WRITE
U 1367, 0000.003C.0180.A800.0000.118A  ;3935  MCT/WRITE.P   ;SYNC:2ND WRITE (BACK TO BACK )
            ;3936 =0 D_K[.C0].LEFT2.SI/MUL*,  ;104 U-SEC WAIT
U 118A, 0878.0039.D300.0800.0000.124B  ;3937   CALL[WAIT.LOOP]
U 118B, 0000.003C.0180.0800.0000.1044  ;3938  NOP
U 1044, 0000.003D.0180.0800.0000.108C  ;3939 =00 CALL[NOINTR]   ;NAKE SURE NO INTERRUPTS PENDING
U 1045, 0000.003D.0180.0800.0000.11BB  ;3940  ERROR1    ;UNEXPECTED INTERRUPT FROM BACK TO BACK
            ;3941      ;  WRITES. MOST LIKELY THE 2ND WRITE
            ;3942      ;  DIDN'T STALL AND BUS FAULT RESULTED =-
U 1046, 0018.0038.1980.0800.0200.1047  ;3943  VA_K[ZERO]   ;READ RESULT OF 1ST WRITE
U 1047, 0000.003C.0180.C800.0000.1048  ;3944  MCT/READ.P   ;SYNC:READ DATA FROM 1ST WRITE
U 1048, 0001.003D.0180.09D8.0000.108C  ;3945 =00 RC[0B]_D,CALL[NOINTR]
U 1049, 0000.003D.0180.0800.0000.11BB  ;3946  ERROR1    ;UNEXPECTED INTERRUPT   =-
U 104A, 0810.0038.0180.0958.0000.104B  ;3947  D_RC[0B]   ;GET DATA READ
U 104B, 0018.0038.C1C0.0800.0000.1368  ;3948  Q_K[.FFFF]   ;GET EXPECTED DATA
U 1368, 001D.0020.0180.0800.0010.1369  ;3949  ALU_D[XOR]Q,CLK.UBCC
U 1369, 0001.213C.0180.09E0.0000.118C  ;3950  Z?,RC[0C]_Q
U 118C, 0000.003D.0180.0800.0000.11BB  ;3951 =0 ERROR1    ;DATA FROM 1ST WRITE IS BAD  =-
U 118D, 0018.0038.7580.0800.0200.136A  ;3952  VA_K[.20]
U 136A, 0000.003C.0180.C800.0000.1050  ;3953  MCT/READ.P   ;SYNC:READ DATA FROM 2ND WRITE
U 1050, 0001.003D.0180.09D8.0000.108C  ;3954 =00 RC[0B]_D,CALL[NOINTR]
U 1051, 0000.003D.0180.0800.0000.11BB  ;3955  ERROR1    ;UNEXPECTED INTERRUPT   =-
U 1052, 0818.0038.C180.0800.0000.1053  ;3956  D_K[.FFFF]
U 1053, 0801.0028.0180.09E0.0000.136B  ;3957  RC[0C]_ALU,D_NOT.D
U 136B, 0011.0020.0180.0958.0010.136C  ;3958  ALU_D[XOR]RC[0B],CLK.UBCC ;COMPARE DATA READ
U 136C, 0000.013C.0180.0800.0000.118E  ;3959  Z?
U 118E, 0000.003D.0180.0800.0000.11BB  ;3960 =0 ERROR1    ;DATA FROM 2ND WRITE IS BAD  =-
U 118F, 0000.003C.0180.0800.0000.1190  ;3961  NOP    ;END SUBTEST
            ;3962
```

```
;3963  .PAGE "TEST 1A2 SBI DISABLE"
;3964 ;*************************************************************
;3965 ;***
;3966 ;
;3967 ; Functional Description:
;3968 ; ----------------------
;3969 ;    This test checks the operation of the DISABLE SBI CYCLE function.
;3970 ;     With SBI cycles disabled,  an SBI parity fault (PD) will be forced. A
;3971 ;     write operation will be attempted followed by a 102 uSec wait, after
;3972 ;     which "no interrupt pending" check will be made.
;3973 ;
;3974 ; Logic Description:
;3975 ; -----------------
;3976 ;  N/A
;3977 ;
;3978 ; Error Logout:
;3979 ; ------------
;3980 ;
;3981 ; See individual error reports.
;3982 ;
;3983 ; Sync Point Description:
;3984 ; ----------------------
;3985 ;
;3986 ;  N/A
;3987 ;
;3988 ;
;3989 ;=============================================================
;3990 ;
;3991 ; Register Map:
;3992 ; ------------
;3993 ;
;3994 ;      RA,RB      Description:
;3995 ;      -----      -----------
;3996 ;
;3997 ;
;3998 ;        F        U-TRAP FLAGS
;3999 ;
;4000 ;
;4001 ;       RC        Description:
;4002 ;       --        -----------
;4003 ; 3 BASE ADDRESS OF MEMORY
;4004 ; D I/O ADDRESS OF  CNFG REG C/D
;4005 ;        E        I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
;4006 ;
;4007 ;        F        NUMBER OF ARGUMENTS TO CONSOLE
;4008 ;
;4009 ;--
;4010 ;*************************************************************
;4011 ;
;4012 =0
U 1190,  0000,003D,0180,0800,0000,11A1  ;4013 TST.17: NEWTST
U 1191,  0003,003C,0180,0998,0000,1192  ;4014  RC[3]_0
U 1192,  0000,003D,0180,0800,0000,121E  ;4015 =0 CALL[CONFIG.MS780E]  ; Find a MS780-E/H on the SBI and
         ;4016       ; ...configure it properly
U 1193,  0000,003C,0180,0800,0000,1194  ;4017  NOP
```

```
          ;4018 ;================================================================================
          ;4019 ;
          ;4020 ;   START THE TESTS
          ;4021 ;
          ;4022 ;================================================================================
          ;4023 ;
U 1194. 0000.003D.0180.0800.0000.11BA  ;4024 =0 ERLOOP    ; Set Uaddress for error loop
U 1195. 0000.003C.0180.0800.0000.1196  ;4025  NOP
U 1196. 0000.003D.0180.0800.0000.124D  ;4026 =0 CALL[INIT]   ; Initialize the SBI
U 1197. 0818.0038.31F8.0800.0000.136D  ;4027  D_K[.40],Q_0   ; Inhibit ECC,set starting
U 136D. 0819.0014.0580.0800.0000.136E  ;4028  D_D+K[.1]   ; ... address to zero
U 136E. 0D10.0038.0180.0970.0200.136F  ;4029  D_DAL.SC,VA_RC[0E]  ; ...
U 136F. 0000.003C.0180.A800.0000.1198  ;4030  MCT/WRITE.P   ; Write it to I/O ADDRESS
          ;4031 =0 D_K[.C0].LEFT2,SI/MUL+,  ; Set up the number of cycles to wait
U 1198. 0878.0039.D300.0800.0000.124B  ;4032   CALL[WAIT.LOOP]  ; Wait
U 1199. 0000.003C.0180.0800.0000.1054  ;4033  NOP    ;
U 1054. 0000.003D.0180.0800.0000.108C  ;4034 =00 CALL[NOINTR]   ; Go check for unexpected interrupts
```

```
          ;4035  .PAGE
          ;4036 ;
          ;4037 ;////////////////////////////////////////////////////////////////////////////////
          ;4038 ;
U 1055. 0000.003D.F580.0800.0084.7064  ;4039  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
          ;4040 ;
          ;4041 ;
          ;4042 ;////////////////////////////////////////////////////////////////////////////////
          ;4043 ;
          ;4044 ; ERROR LOGOUT:
          ;4045 ;
          ;4046 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
          ;4047 ;  NOT USED
          ;4048 ;  NOT USED
          ;4049 ;  NOT USED
          ;4050 ;  NOT USED
          ;4051 ;  NOT USED
          ;4052 ;  ID VECTOR REGISTER
          ;4053 ;  ID SBI ERROR.REGISTER
          ;4054 ;  ID FAULT REGISTER
          ;4055 ;  INTERRUPT FLAG (TRUE=1, FALSE=0)
          ;4056 ;
          ;4057 ;////////////////////////////////////////////////////////////////////////////////
          ;4058 ;
U 1056. 0F00.003C.8D80.0800.0084.7370  ;4059  D_0,SC_K[.1F]   ; Set Force P0 Fault
U 1370. 0801.001C.0180.0800.0000.1371  ;4060 = D_D.ORNOT.MASK
U 1371. 0000.003C.8580.0800.0084.7372  ;4061  SC_K[.C]   ; Set Disable SBI Cycles
U 1372. 0801.001C.0180.0800.0000.1373  ;4062  D_D.ORNOT.MASK
U 1373. 0000.003C.75F0.2C00.0000.1374  ;4063  Q_ID[MAINT]   ; Get cache off Bits
U 1374. 081D.0030.0180.0800.0000.1375  ;4064  D_D[OR]Q   ; Generate the complete constant
U 1375. 0010.0038.7580.3D18.0200.1376  ;4065  ID[MAINT]_D,VA_RC[3]  ; Write the maintance reg,set address
U 1376. 0000.003C.0180.A800.0000.119A  ;4066  MCT/WRITE.P   ; Start the write operation
          ;4067 =0 D_K[.C0].LEFT2,SI/MUL*.  ; Set up the number of cycles to wait
U 119A. 0878.0039.D300.0800.0000.124B  ;4068   CALL[WAIT.LOOP]  ; Wait 102 u-sec
U 119B. 0000.003C.0180.0800.0000.1060  ;4069  NOP
U 1060. 0000.003D.0180.0800.0000.108C  ;4070 =00 CALL[NOINTR]   ; Check for no SBI Faults
```

```
            ;4071   .PAGE
            ;4072 ;
            ;4073 ;////////////////////////////////////////////////////////////////////////////////
            ;4074 ;
U 1061, 0000.003D.F580.0800.0084.7064  ;4075  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
            ;4076 ;
            ;4077 ;
            ;4078 ;////////////////////////////////////////////////////////////////////////////////
            ;4079 ;
            ;4080 ; ERROR LOGOUT:
            ;4081 ;
            ;4082 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4083 ;   NOT USED
            ;4084 ;   NOT USED
            ;4085 ;   NOT USED
            ;4086 ;   NOT USED
            ;4087 ;   NOT USED
            ;4088 ;   ID VECTOR REGISTER
            ;4089 ;   ID SBI.ERROR REGISTER
            ;4090 ;   ID FAULT REGISTER
            ;4091 ;   INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;4092 ;
            ;4093 ;////////////////////////////////////////////////////////////////////////////////
            ;4094 ;
            ;4095
U 1062, 0000.003C.0180.0800.[illegible]0.119C  ;4096  NOP
            ;4097 =
            ;4098
```

```
            ;4099  .PAGE "TEST 1A3 RESERVED"
            ;4100 =0
U 119C, 0000.003D.0180.0800.0000.11A1  ;4101 T1A3: NEWTST
U 119D, 0000.003C.0180.0800.0000.119E  ;4102  NOP
            ;4103
```

```
            ;4104  .PAGE "TEST 1A4 RESERVED"
            ;4105 =0
U 119E, 0000,003D,0180,0800,0000,11A1  ;4106 T1A4: NEWTST
U 119F, 0000,003C,0180,0800,0000,1377  ;4107  NOP
            ;4108
```

```
          ;4109  .PAGE "DUMMY NEWTST"
U 1377, 0000,003D,0180,0800,0000,11A1  ;4110 DUM: NEWTST
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 2556= 2557= 2558= 2560=
U 1008  2188= 2189= 2190= 2191= 2192= 2193= 2194= 2195=
U 1010  2605= 2606= 2607= 2609= 2656= 2657= 2658= 1875
U 1018  2663= 2664= 2665= 1876 2669= 2670= 2671= 1877
U 1020  2675= 2678= 2680= 2681= 3116= 3117= 3118= 3119=
U 1028  3231= 3232= 3233= 1878 3235= 3236= 3237= 1879
U 1030  3246= 3247= 3257= 3258= 3288= 3289= 3293= 3294=
U 1038  3363= 3364= 3365= 3366= 3386= 3387= 3394= 3395=
U 1040  3607= 3608= 3609= 3610= 3939= 3940= 3943= 3944=
U 1048  3945= 3946= 3947= 3948= 1898= 1899= 2003: 1880
U 1050  3954= 3955= 3956= 3957= 4034= 4039= 4059= 1900
U 1058  1958= 1960= 2029: 1901 1971= 1972= 2054: 1902
U 1060  4070= 4075= 4096= 1903 2021= 2022= 2046= 2047=
U 1068  2167= 2168= 2176= 2177= 2178= 2180= 2183= 2184=
U 1070  2225= 2226= 2227= 2228= 2235= 2236= 2242= 2244=
U 1078  2247= 2248= 2267= 2268= 2270= 2271= 2363= 2364=
U 1080  2367= 2368= 2379= 2380= 2383= 2384= 2445= 2446=
U 1088  2468= 2469= 2511= 2512= 2726= 2727= 2730= 2731=
U 1090  2733= 2734= 2738= 2739= 2822= 2823= 2840= 2841=
U 1098  2843= 2844= 2846= 2847= 2859= 2860= 2861= 2863=
U 10A0  2864= 2865= 2866= 2867= 2868= 2869= 2870= 2871=
U 10A8  2872= 2873= 2875= 2876= 2910= 2911= 2922= 2923=
U 10B0  3074= 3075= 3076= 3078= 3089= 3090= 3091= 3092=
U 10B8  3101= 3102= 3113= 3114= 3130= 3131= 3145= 3146=
U 10C0  3169= 3170= 3171= 3172= 3219= 3220= 3221= 3222=
U 10C8  3223= 3224= 3225= 3226= 3229= 3230= 3239= 3240=
U 10D0  3244= 3245= 3263= 3264= 3268= 3269= 3286= 3287=
U 10D8  3299= 3300= 3304= 3305= 3353= 3354= 3355= 3356=
U 10E0  3357= 3358= 3361= 3362= 3368= 3369= 3375= 3376=
U 10E8  3377= 3378= 3379= 3380= 3384= 3385= 3400= 3401=
U 10F0  3404= 3405= 3458= 3459= 3460= 3461= 3462= 3463=
U 10F8  3464= 3465= 3474= 3478= 3479= 3480= 3487= 3488=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1904 3491= 3495= 1870: 1871: 1872: 1873:
U 1110  3498= 3499= 3506= 3510= 3516= 3520= 3524= 3528=
U 1118  3532= 3537= 3542= 3546= 3549= 3550= 3588= 3589=
U 1120  3590= 3591= 3592= 3593= 3602= 3603= 3604= 3606=
U 1128  3615= 3616= 3620= 3621= 3632= 3637= 3638= 3639=
U 1130  3644= 3645= 3648= 3649= 3658= 3665= 3699= 3700=
U 1138  3701= 3702= 3703= 3704= 3707= 3708= 3709= 3711=
U 1140  3713= 3714= 3715= 3716= 3724= 3725= 3729= 3730=
U 1148  3733= 3734= 3738= 3740= 3743= 3744= 3762= 3763=
U 1150  3764= 3765= 3766= 3767= 1905 2076: 3769= 3770=
U 1158  3773= 3774= 3778= 3779= 3795= 3796= 3797= 3798=
U 1160  3799= 3800= 3801= 3802= 3805= 3807= 3809= 3810=
U 1168  3815= 3816= 3826= 3827= 3831= 3833= 3840= 3841=
U 1170  3844= 3845= 3848= 3850= 3868= 3869= 3870= 3871=
U 1178  3872= 3873= 3874= 3875= 3878= 3879= 3886= 3888=
U 1180  3919= 3920= 3921= 3922= 3923= 3924= 3927= 3928=
U 1188  3930= 3931= 3937= 3938= 3951= 3952= 3960= 3961=
U 1190  4013= 4014= 4015= 4017= 4024= 4025= 4026= 4027=
U 1198  4032= 4033= 4068= 4069= 4101= 4102= 4106= 4107=
```

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0 1906 1947 1948 1949 1950 1951 1952 1953
U 11A8 1954 1955 1956 1957 1961 1962 1963 1964
U 11B0 1965 1966 1967 1968 1969 1970 1991 1992
U 11B8 1993 1994 1998 2001 2002 2027 2028 2099
U 11C0 2108 2109 2118 2119 2127 2128 2129 2169
U 11C8 2170 2171 2172 2181 2182 2186 2187 2200
U 11D0 2201 2203 2204 2205 2207 2212 2213 2214
U 11D8 2216 2221 2223 2224 2232 2233 2237 2238
U 11E0 2239 2240 2241 2249 2250 2251 2253 2254
U 11E8 2255 2256 2257 2258 2263 2265 2266 2269
U 11F0 2294 2295 2296 2297 2326 2327 2329 2330
U 11F8 2331 2359 2361 2362 2366 2370 2371 2372
U 1200 2373 2375 2376 2377 2378 2381 2382 2411
U 1208 2412 2413 2438 2439 2440 2441 2443 2444
U 1210 2470 2471 2472 2473 2500 2501 2502 2503
U 1218 2504 2505 2506 2507 2509 2510 2654 2655
U 1220 2682 2697 2698 2699 2700 2701 2702 2703
U 1228 2728 2729 2735 2736 2737 2740 2741 2742
U 1230 2743 2744 2745 2746 2747 2764 2765 2766
U 1238 2767 2768 2769 2770 2771 2772 2773 2774
U 1240 2775 2776 2777 2778 2779 2781 2782 2783
U 1248 2784 2785 2790 2817 2838 2858 2877 2878
U 1250 2903 2904 2905 2906 2907 2908 2909 2916
U 1258 2917 2918 2919 2920 2921 2932 2933 2934
U 1260 2942 2943 2944 2945 2953 2954 2955 2956
U 1268 2964 2965 2966 2967 2974 2975 2976 2977
U 1270 3079 3080 3081 3082 3083 3084 3088 3093
U 1278 3094 3095 3096 3097 3098 3099 3100 3104
U 1280 3105 3106 3107 3108 3109 3110 3111 3115
U 1288 3120 3121 3122 3123 3124 3125 3126 3127
U 1290 3128 3129 3132 3133 3134 3135 3136 3137
U 1298 3138 3139 3140 3141 3142 3143 3144 3147
U 12A0 3148 3149 3150 3151 3152 3153 3154 3155
U 12A8 3156 3157 2077: 3158 3159 3160 3161 3162
U 12B0 3163 3164 3165 3166 3167 3168 3173 3227
U 12B8 3234 3238 3259 3260 3261 3262 3265 3266
U 12C0 3267 3281 3282 3283 3284 3295 3296 3297
U 12C8 3298 3301 3302 3303 3312 3359 3367 3370
U 12D0 3371 3372 3373 3374 3381 3382 3396 3397
U 12D8 3398 3399 3402 3403 3412 3466 3467 3468
U 12E0 3469 3470 3471 3472 3473 3484 3485 3486
U 12E8 3489 3490 3496 3497 3500 3501 3502 3503
U 12F0 3504 3505 3511 3512 3513 3514 3515 3521
U 12F8 3522 3523 3529 3530 3531 3538 3539 3540
U 1300 3541 3547 3548 3594 3595 3596 3597 3598
U 1308 3599 3600 3611 3612 3613 3614 3617 3618
U 1310 3619 3622 3623 3624 3625 3626 3627 3628
U 1318 3629 3630 3631 3640 3641 3642 3643 3646
U 1320 3647 3650 3651 3652 3653 3654 3655 3656
U 1328 3657 3705 3706 3712 3717 3718 3719 3720
U 1330 3721 3722 3723 3726 3727 3728 3735 3736
U 1338 3737 3741 3742 3768 3771 3772 3775 3776
U 1340 3777 3780 3803 3804 3808 3812 3813 3814

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3817 3819 3820 3821 3822 3823 3824 3825
U 1350  3828 3829 3830 3834 3836 3837 3838 3839
U 1358  3846 3847 3876 3877 3880 3881 3882 3883
U 1360  3884 3885 3925 3929 3932 3933 3934 3935
U 1368  3949 3950 3953 3958 3959 4028 4029 4030
U 1370  4060 4061 4062 4063 4064 4065 4066 4110
U 1378 - 13EF Unused
U 13F0  2060: 2061: 2062: 2063: 2064: 2065: 2066: 2067:
U 13F8  2068: 2069: 2070: 2071: 2072: 2073: 2074: 2075:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR44	904
Used	904
Remaining	

Total microwords used in memory U: 904
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0658
ESKAR44.MCR                    MICRO2  1P(00)    26-JUL-85  17:12:03     Page  12


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR44.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 0659
ESKAR45.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2173 Diagnostic Specific Subroutines
; 2174   Select Controller
; 2212   START TEST
; 2421   SELECT LOWER CONTROLLER
; 2469   SELECT UPPER CONTROLLER
; 2517   Set Diagnostic Mode Routine
; 2546   SBI Unjam Routine
; 2595   RESET SUBTEST NUMBER
; 2617   Initialize the SBI
; 2644   Disable Cache Routine
; 2665   Enable Cache
; 2686   Wait Loop Routine
; 2719   Check for Interrupt Routine
; 2766   CHECK UTRAP
; 2798   Set Trap Mask
; 2826   FIND MS780-E MEMORY
; 2968   Generate IO Address
; 2993   Set Starting Address
; 3026   ENABLE NEXT MS780-E
; 3078   MS780-E/H CNFG REG CLEANUP
; 3147 TEST 1A5 MICRO SEQUENCER PARITY ERROR
; 3345 TEST 1A6 D BUS PARITY CHECKING/GENERATION
; 3656 TEST 1A7 ROM READ/WRITE ACCESS
; 4143 TEST 1A8 BYTE WRITE TO MEMORY TEST
; 4694 TEST 1A9 UNALIGNED READ/WRITE FROM/TO MEMORY
; 5284 TEST 1AA RESERVED
; 5289 TEST 1AB RESERVED
; 5294 TEST 1AC RESERVED
; 5299 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR45
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
;1841  .PAGE
;1842
;1843  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844  .TOC "   Micro Trap Handler"
;1845 ;++
;1846 ; FUNCTIONAL DESCRIPTION:
;1847 ;
;1848 ; This routine is responsible for 'catching' micro-traps
;1849 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850 ; contains the Trap-Expected Mask. If the specified bit is set in
;1851 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852 ; an error call is made.
;1853 ;
;1854 ; INPUT PARAMETERS:
;1855 ;
;1856 ; R[0F] - Expected Trap Mask
;1857 ;  Trap Code Function
;1858 ;  --------      --------
;1859 ;       0  System Init
;1860 ;       1  Data Unaligned
;1861 ;       2  Cross Page
;1862 ;       3  TB Modify
;1863 ;       4  TB Protection
;1864 ;       6  TB Miss
;1865 ;       7  TB Parity
;1866 ;       8  Cache Parity
;1867 ;       9  Reserved Floating Operand
;1868 ;       C  Read Data Substitute
;1869 ;       D  Time out
;1870 ;       F  Control Store Parity Error
;1871 ;      10  Odd Address
;1872 ;
;1873 ; OUTPUT PARAMETERS:
;1874 ;
;1875 ; R[0F] - Mask bit is cleared.
;1876 ;
;1877 ; DATA: Micro PC of Micro Trap
;1878 ;  Trap Code (See Above)
;1879 ;  Fault Status Register
;1880 ;  SBI Error Register
;1881 ;  Timeout Address Register
;1882 ;  Maintenance Register
;1883 ;  Cache Parity Register
;1884 ;  TB Error Register 1
;1885 ;  TB Error Register 0
;1886 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C. 0000.003C.8580.0800.0084.7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D. 0000.003C.8980.0800.0084.7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E. 0000.003C.6580.0800.0084.7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F. 0000.003C.6180.0800.0084.7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1900
U 1001. 0000.003C.81F0.2C00.0000.1033  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 1033. 0C00.003C.01E0.0800.0000.1043  ;1902  q_d,d_q   ; Setup to put it back on stack
U 1043. 0C00.003C.81E0.3C00.0000.1047  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1047. 0000.003C.01C0.0A78.0000.104B  ;1904  q_r[0f]   ; Get the Expected Trap Mask
          ;1905  alu_q.andnot.mask,
U 104B. 0001.2024.0180.0800.0010.104F  ;1906   clk.ubcc ; Was trap expected?
U 104F. 0000.013C.0180.0800.0000.1002  ;1907  z?
          ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1909      alu_q.and.mask, ; ...
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1910       return0  ; ...
U 1003. 0018.0038.1D80.09E8.0000.104C  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 104C. 0000.003D.D980.0800.0084.71D8  ;1912 =0 setrcf[.9]  ; Set output count
U 104D. 0000.003C.49F0.2C00.0000.1053  ;1913  q_id[tber0]  ; Output all the error registers
U 1053. 0001.203C.4DF0.2DB0.0000.105B  ;1914  rc[6]_q,q_id[tber1] ; ...
U 105B. 0001.203C.65F0.2DB8.0000.105F  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 105F. 0001.203C.6DF0.2DD8.0000.106F  ;1916  rc[0b]_q,q_id[fault] ; ...
U 106F. 0001.203C.75F0.2DE0.0000.107F  ;1917  rc[0c]_q,q_id[maint] ; ...
U 107F. 0001.203C.79F0.2DC8.0000.1083  ;1918  rc[9]_q,q_id[parity] ; ...
U 1083. 0001.203C.69F0.2DC0.0000.108B  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 108B. 0001.203C.81F0.2DD0.0000.1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
          ;1921 1000: ERROR1[.C],  ;
U 1000. 0001.203D.8580.09F0.0084.70A4  ;1922   rc[0e]_q ; Report the error
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "   Newtest Routine"
                ;1926 ;++
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to     0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 109B, 0000,003C,65F8,0800,0084,7109  ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1109, 0818,0038,8D80,0800,0000,11C0  ;1955  d_k[.1f]  ; D=1F
U 11C0, 0D00,003C,0180,0800,0000,11C1  ;1956  d_dal.sc  ; D=001F0000
U 11C1, 0000,003C,3D80,3C00,0000,11C2  ;1957  id[psl]_d  ; psl = 001F0000
U 11C2, 0818,0038,0580,0800,0000,11C3  ;1958  d_k[.1]  ; Clear the CES register
U 11C3, 0D00,003C,0180,0800,0000,11C4  ;1959  d_dal.sc  ; D = 00010000
U 11C4, 0F00,003C,3180,3C00,0000,11C5  ;1960  id[ces]_d,d_0  ; ces = 00010000
U 11C5, 0000,003C,5D80,3C00,0000,11C6  ;1961  id[acc.cs]_d  ; acc.cs = 0
U 11C6, 0000,003C,3980,3C00,0000,11C7  ;1962  id[sir]_d  ; sir = 0
U 11C7, 0000,003C,2980,3C00,0000,11C8  ;1963  id[clk.cs]_d  ; clk.cs = 0
U 11C8, 0000,003C,4980,3C00,0000,1058  ;1964  id[tber0]_d  ; tber0 = 0
U 1058, 0000,003D,0180,0800,0000,11F4  ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1966  intrpt.strobe,  ; Strobe interrupts
U 1059, 0818,0038,C180,0800,4000,11C9  ;1967      d_k[.ffff]  ; Setup to clear SBI error register and
U 11C9, 0801,0028,6580,3C00,0000,11CA  ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 11CA, 0F00,003C,6D80,3C00,0000,11CB  ;1969  id[fault]_d,d_0  ; ...
U 11CB, 0000,003C,6D80,3C00,0000,11CC  ;1970  id[fault]_d  ; fault = 0
U 11CC, 0000,003C,7580,3C00,0000,11CD  ;1971  id[maint]_d  ; MAINT = 0
U 11CD, 0000,003C,6580,3C00,0000,11CE  ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 11CE, 0000,003C,7180,3C00,0000,11CF  ;1973  id[comp]_d  ; comp reg = 0
U 11CF, 0000,003C,E580,3C00,0000,11D0  ;1974  ID[39]_d  ; Clear code for subroutine START.TEST
U 11D0, 0001,003C,0180,09F0,0000,11D1  ;1975  rc[0e]_d  ;
U 11D1, 0001,003C,0180,0AF8,0000,11D2  ;1976  r[0f]_d  ; Clear expected trap mask
U 11D2, 0001,003C,0180,09F8,0000,105C  ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,11FA  ;1978 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1980  .PAGE
            ;1981  .TOC "   Error Loop Routine"
            ;1982 ;++
            ;1983 ; FUNCTIONAL DESCRIPTION:
            ;1984 ;
            ;1985 ; This routine is called with the "ERLOOP" macro. The
            ;1986 ; routine calls the Micro Monitor to setup an error loop.
            ;1987 ;
            ;1988 ; CALLING CONSTRAINT:
            ;1989 ;
            ;1990 ; =0
            ;1991 ;
            ;1992 ; SIDE EFFECTS:
            ;1993 ;
            ;1994 ; D Reg - Destroyed
            ;1995 ;--
U 11D3, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 11D4, 0810,0038,0180,0978,0000,11D5  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 11D5, 0819,0034,4DB0,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,11B0,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;**
            ;2025 ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 10A4, 0000,003D,0180,0800,0000,11D8  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10A5, 0000,003C,0180,0800,0000,11D4  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "   Error 2 Routine"
            ;2045 ;++
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 11D6, 0810,0038,0180,0978,0000,11D7  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 11D7, 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "    ERROR 2 WITH PARAMETER"
            ;2071 ;++
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 10A6, 0000,003D,0180,0800,0000,11D8  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10A7, 0000,003C,0180,0800,0000,11D6  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
            ;2088  .PAGE
            ;2089  .TOC "   Micro-Monitor Call"
            ;2090 ;*
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;++
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2111 13F0: nop
U 13F1, 0000.003C.0180.0800.0000.13F2  ;2112 13F1: nop
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2113 13F2: nop
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2114 13F3: nop
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2115 13F4: nop
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2116 13F5: nop
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2117 13F6: nop
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2118 13F7: nop
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2119 13F8: nop
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2120 13F9: nop
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2121 13FA: nop
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2122 13FB: nop
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2123 13FC: nop
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2124 13FD: nop
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2125 13FE: nop
U 13FF, 0000.003C.0180.0800.0000.12AA  ;2126 13FF: nop
U 12AA, 0000.003C.0180.0800.0000.11D8  ;2127 12AA: nop ; This location is permanently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;++
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 11D8, 0010,0038,01C0,0978,0000,11D9  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 11D9, 0019,2034,4DC0,0800,0000,11DA  ;2150  q_q.and.k[.ff00] ; ...
U 11DA, 0019,2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "   Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 11DB, 0810,0038,4D80,0978,0000,11DC  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 11DC, 0019,4016,4D80,09F8,0000,0001  ;2171       word,return1 ; (Subtest number is negative)
```

```
            ;2172  .PAGE
            ;2173  .TOC " Diagnostic Specific Subroutines"
            ;2174  .TOC "   Select Controller"
            ;2175 ;+*
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine is used to put the memory system into the
            ;2179 ; specified configuration with respect to controller select.
            ;2180 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2181 ; a RETURN2 is made.
            ;2182 ;
            ;2183 ; CALLING CONSTRAINT:
            ;2184 ;
            ;2185 ; =00
            ;2186 ;
            ;2187 ; INPUTS:
            ;2188 ;
            ;2189 ; d - Contains value to write to bits<2:0> of Register A
            ;2190 ; rc[0e] - Address of Register A
            ;2191 ;
            ;2192 ; SIDE EFFECTS:
            ;2193 ;
            ;2194 ; sc,d,va are destroyed
            ;2195 ;--
            ;2196 SELECT.CNTRLR:
U 11DD, 0010.0038.0180.0970.0284.71DE  ;2197  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11DE, 0801.001C.0180.0800.0000.11DF  ;2198  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11DF, 0000.003C.0180.A800.0000.11E0  ;2199  cache.p_d[long]  ; Write the configuration Register
U 11E0, 0818.0038.5D80.0800.0000.11E1  ;2200  d_k[.7]  ; Get Misconfiguration bits
U 11E1, 0000.003C.61F8.0800.0084.71E2  ;2201  sc_k[.F],q_0  ; ...
U 11E2, 0D00.003C.0180.0800.0000.11E3  ;2202  d_dal.sc  ; ... D = x38000
U 11E3, 0000.003C.01E0.0800.0000.11E4  ;2203  q_d  ; Save mask
U 11E4, 0000.003C.0180.C800.0000.11E5  ;2204  d[long]_cache.p  ; Read CNFG A Reg and
            ;2205  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11E5, 001D.2034.0180.0800.0010.11E6  ;2206  clk.ubcc ; ...
U 11E6, 0000.013C.0180.0800.0000.10A8  ;2207  z?  ; Branch if no
U 10A8, 0000.003E.0180.0800.0000.0001  ;2208 =0 RETURN1  ; Bad configuration
U 10A9, 0000.003E.0180.0800.0000.0002  ;2209  RETURN2  ; Configuration ok
            ;2210
```

```
;2211  .PAGE
;2212  .TOC "   START TEST"
;2213 ;*************************************************************
;2214 ;++
;2215 ;
;2216 ; Functional Description:
;2217 ; ---------------------
;2218 ;
;2219 ;         This subroutine will be called by all tests that  check  the
;2220 ;    controllers,  or  those  tests  that  have  to switch between the
;2221 ;    controllers when executing.   Its  main  functions  are:   select
;2222 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2223 ;    test that called it, call out an error  if  neither  one  of  the
;2224 ;    controllers can be configured properly, if all the controllers on
;2225 ;    a MS780-E/H were configured and tested, it should select the next
;2226 ;    MS780-E/H on the SBI and repeat the above sequence.
;2227 ;    A test making use of this  subroutine  should  be  configured  as
;2228 ;    follows:
;2229 ;
;2230 ;
;2231 ;        Begin TEST.XX
;2232 ;         :
;2233 ;         :   Call NEWTST
;2234 ;         :   Call FIND.MS780E
;2235 ;         :   IF No MS780-E's on the SBI
;2236 ;         :     THEN
;2237 ;         :         Skip to End of TEST.XX
;2238 ;         :     ELSE
;2239 ;         :
;2240 ;         : RESTART.TEST.XX:
;2241 ;         :
;2242 ;         :   Call START TEST
;2243 ;         :   IF Return1 from START.TEST
;2244 ;         :     THEN
;2245 ;         :         Skip to End of TEST.XX
;2246 ;         :     ELSE.
;2247 ;         :
;2248 ;         :
;2249 ;         :
;2250 ;         :
;2251 ;         :   Body of TEST.XX
;2252 ;         :
;2253 ;         :
;2254 ;
;2255 ;         :   Jump RESTART.TEST.XX
;2256 ;         :
;2257 ;        End TEST.XX
;2258 ;
;2259 ;
;2260 ;          This subroutine makes use of a pointer  in  ID  Register  39
;2261 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2262 ;    should be passed when the subroutine is called a second, third or
;2263 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2264 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2265 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2266 ;    Register 39 can be interpreted as follows:
;2267 ;
;2268 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2269 ;               called the first time.  When this code is encountered
;2270 ;               this subroutine will try to select the lower controller.
;2271 ;               If the lower controller is misconfigured, the pointer in
;2272 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2273 ;               is re-entered.
;2274 ;               If, however there is no misconfiguration, the value in
;2275 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2276 ;               calling test.
;2277 ;
;2278 ;     b01   -   This value signifies that an attempt at configuring the
;2279 ;               lower controller failed, and the subroutine will attempt
;2280 ;               to select the upper controller.
;2281 ;               If the upper controller is also misconfigured execution
;2282 ;               stops with a call to ERROR1.
;2283 ;               If there is no misconfiguration on this controller, then
;2284 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2285 ;               made to the calling test.
;2286 ;
;2287 ;     b10   -   This value signifies that the lower controller was selected
;2288 ;               previously and the test was executed once.  If the
;2289 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2290 ;               39 is loaded with b11 and an attempt is made to select the
;2291 ;               upper controller.
;2292 ;               If there is a misconfiguration on this controller, this
;2293 ;               subroutine is just re-entered.
;2294 ;               Otherwise, a Return2 will be made to the calling test.
;2295 ;
;2296 ;     b11   -   This code identifies the cases in which the test has
;2297 ;               been executed at least once (i.e., at least one of the
;2298 ;               controllers on the MS780-E unit under test could be
;2299 ;               configured properly).  When this code is detected the
;2300 ;               subroutine clears ID Reg 39 and makes a call to
;2301 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2302 ;               tested then the subroutine executes a Return1.
;2303 ;               Otherwise the subroutine is re-entered.
;2304 ;
;2305 ;
;2306 ; Calling Constraint: =00
;2307 ; ------------------
;2308 ;
;2309 ;
;2310 ; Inputs:
;2311 ; ------
;2312 ;
;2313 ; ID Register 39 must be cleared by NEWTST
;2314 ;
;2315 ;
;2316 ; Outputs:
;2317 ; -------
;2318 ;
;2319 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2320 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2321 ;         to be tested
            ;2322 ;
            ;2323 ; Side Effects:
            ;2324 ; ------------
            ;2325 ;
            ;2326 ; Contents of the following regiters will be destroyed:
            ;2327 ;
            ;2328 ; D, Q
            ;2329 ;
            ;2330 ;--
            ;2331 ;*************************************************************
            ;2332 =0
            ;2333 START.TEST:
U 10AA, 0000,003D,0180,0800,0000,11F5  ;2334  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 10AB, 0000,003C,0180,0800,0000,10AC  ;2335  NOP    ; ...tests that have more than one
            ;2336 =0      ; ...subtest.)
U 10AC, 0000,003D,0180,0800,0000,11D3  ;2337  ERLOOP    ; Set up the error loop for the
            ;2338       ; ...eventuality that the MS780-E/H
            ;2339       ; ...currently under test has no
            ;2340       ; ...Controllers
            ;2341 RE.ENTRY:    ; Re entry point for this routine
U 10AD, 0000,003C,E5F0,2C00,0000,11E7  ;2342  Q_ID[39]    ; Get the code
U 11E7, 0C00,003C,0180,0800,0000,11E8  ;2343  D_Q    ; Put it in the D Register
U 11E8, 0000,193C,0180,0800,0000,100C  ;2344  DI-0?    ; Branch on the code
U 100C, 0000,003C,0180,0800,0000,1008  ;2345 =1100 J/STRT.CASE00    ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1010  ;2346  J/STRT.CASE01    ; Code is 01
U 100E, 0000,003C,0180,0800,0000,11ED  ;2347  J/STRT.CASE10    ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1017  ;2348  J/STRT.CASE11    ; Code is 11
            ;2349 ;
            ;2350 ;
            ;2351 ; At this point the code in ID[39] was 00
            ;2352 ;
            ;2353 ;
            ;2354 =00
            ;2355 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1024  ;2356  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2357  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,11E9  ;2358   D_K[.1]    ; Set up the code for re entry to this
            ;2359       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2360  D_K[.2]    ; Set up the code for a next call to
            ;2361       ; ...START.TEST indicating that the
            ;2362       ; ...Lower Controller was configured
            ;2363       ; ...( 10 )
            ;2364  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2365   RETURN2    ; Let the test know that the Lower
            ;2366       ; ...has been configured
            ;2367 STRT.LOW.MIS:
            ;2368  ID[39]_D,    ; Save code in ID[39] signifing that
U 11E9, 0000,003C,E580,3C00,0000,10AD  ;2369   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2370       ; ...and re enter this subroutine
            ;2371
            ;2372 ;
            ;2373 ;
            ;2374 ; At this point the code is 01
            ;2375 ;
```

```
            ;2376 ;
            ;2377 =00
            ;2378 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1028  ;2379  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2380  J/STRT.UPR.MIS.   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,11EA  ;2381   D_0   ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2382  D_K[.3]    ; Upper controller was configured
            ;2383       ; ...thus the code must be changed
            ;2384       ; ...accordingly ( 11 )
            ;2385  ID[39]_D,   ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2386   RETURN2   ; Let the test know that the Upper
            ;2387       ; ...Controller has been configured
            ;2388 STRT.UPR.MIS:
U 11EA, 0000,003C,E580,3C00,0000,11EB  ;2389  ID[39]_D   ; Save the Code ( 00 )
U 11EB, 0018,0038,0D80,09E8,0000,11EC  ;2390  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 11EC, 0000,003D,0980,0800,0084,70A4  ;2391  ERROR1[.2]   ; Flag the error.
            ;2392 ;
            ;2393 ;
            ;2394 ; At this point the code is 10
            ;2395 ;
            ;2396 ;
            ;2397 STRT.CASE10:
U 11ED, 0818,0038,0D80,0800,0000,1014  ;2398  D_K[.3]    ; Set the code to 11
            ;2399
            ;2400
            ;2401 =00 ID[39]_D,   ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1028  ;2402   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,10AD  ;2403  J/RE.ENTRY   ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2404  RETURN2    ; Upper Controller configured
            ;2405       ; ...let the test know it
            ;2406 ;
            ;2407 ;
            ;2408 ; At this point the code is 11
            ;2409 ;
            ;2410 ;
            ;2411 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1020  ;2412  D_0    ; Clear the code
            ;2413 =00 ID[39]_D,   ; Save the code
U 1020, 0000,003D,E580,3C00,0000,124B  ;2414   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2415       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,10AD  ;2416  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2417       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2418  RETURN1    ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1024  ;2419  NOP
```

```
;2420  .PAGE
;2421  .TOC "   SELECT LOWER CONTROLLER"
;2422 ;*************************************************************
;2423 ;++
;2424 ;
;2425 ; Functional Description:
;2426 ; ----------------------
;2427 ;
;2428 ; This subroutine can be called to select the lower
;2429 ; Controller on a MS780-E/H.
;2430 ; If the Lower controller is misconfigured then a
;2431 ; RETURN1 will be made.  Otherwise a RETURN2
;2432 ;
;2433 ; Calling Constraint: =00
;2434 ; ------------------
;2435 ;
;2436 ;
;2437 ; Inputs:
;2438 ; ------
;2439 ;
;2440 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2441 ;
;2442 ; Outputs:
;2443 ; -------
;2444 ;
;2445 ; RC[0D] will have the address of CNFG Register C
;2446 ;        ( 2000x008 )
;2447 ;
;2448 ; Side Effects:
;2449 ; ------------
;2450 ;
;2451 ; Contents of the following registers will lost:
;2452 ;
;2453 ; D
;2454 ;
;2455 ; The Lower controller on the MS780-E/H will be configured
;2456 ; when the subroutine is exitted with a RETURN2
;2457 ;--
;2458 ;*************************************************************
;2459 =00
;2460 SELECT.LOWER:
;2461   D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,11DD  ;2462   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2463  RETURN1     ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2464  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2465  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2466   RETURN2    ; Let caller know that the controller
;2467       ; ...was configured properly
```

```
;2468  .PAGE
;2469  .TOC "   SELECT UPPER CONTROLLER"
;2470 ;*************************************************************
;2471 ;**
;2472 ;
;2473 ; Functional Description:
;2474 ; ----------------------
;2475 ;
;2476 ; This subroutine can be called to select the upper
;2477 ; Controller on a MS780-E/H.
;2478 ; If the Upper controller is misconfigured then a
;2479 ; RETURN1 will be made.  Otherwise a RETURN2
;2480 ;
;2481 ; Calling Constraint: =00
;2482 ; ------------------
;2483 ;
;2484 ;
;2485 ; Inputs:
;2486 ; ------
;2487 ;
;2488 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2489 ;
;2490 ; Outputs:
;2491 ; -------
;2492 ;
;2493 ; RC[0D] will have the address of CNFG Register D
;2494 ;         ( 2000x010 )
;2495 ;
;2496 ; Side Effects:
;2497 ; ------------
;2498 ;
;2499 ; Contents of the following registers will lost:
;2500 ;
;2501 ; D
;2502 ;
;2503 ; The Upper controller on the MS780-E/H will be configured
;2504 ; when the subroutine is exitted with a RETURN2
;2505 ;--
;2506 ;*************************************************************
;2507 =00
;2508 SELECT.UPPER:
;2509   D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,11DD  ;2510    CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2511   RETURN1     ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2512   D_RC[0E]    ; Get MS780-E/H I/O Base address
;2513   RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2514    RETURN2    ; Let caller know that the controller
;2515        ; ...was configured properly
```

```
            ;2516  .PAGE
            ;2517  .TOC "   Set Diagnostic Mode Routine"
            ;2518 ;++
            ;2519 ; FUNCTIONAL DESCRIPTION:
            ;2520 ;
            ;2521 ; This routine is used to set the specified diagnostic mode
            ;2522 ; in Register B. Other bits in the register remain unchanged.
            ;2523 ;
            ;2524 ; CALLING CONSTRAINT:
            ;2525 ;
            ;2526 ; =0
            ;2527 ;
            ;2528 ; INPUTS:
            ;2529 ;
            ;2530 ; d - Contains the mode to set in bits<1:0>
            ;2531 ; rc[0e] - Contains base address of controller
            ;2532 ;
            ;2533 ; SIDE EFFECTS:
            ;2534 ;
            ;2535 ; d,va,sc are destroyed
            ;2536 ;--
            ;2537 SET_DIAG_MODE:
U 11EE, 0010,0038,D9F8,0970,0284,71EF  ;2538  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11EF, 0D00,003C,0180,0803,0000,11F0  ;2539  va_va+4,d_dal.sc ; Shift specified mode into position
U 11F0, 0000,003C,01E0,0800,0000,11F1  ;2540  q_d    ; Save the diagnostic mode bits
U 11F1, 0000,003C,0180,C800,0000,11F2  ;2541  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11F2, 081D,0030,0180,0800,0000,11F3  ;2542  d_d.or.q  ; Set the diagnostic mode bits
U 11F3, 0000,003E,0180,A800,0000,0001  ;2543  cache.p_d[long],return1 ; Set the specified mode and return
            ;2544
```

```
            ;2545  .PAGE
            ;2546  .TOC "   SBI Unjam Routine"
            ;2547 ;++
            ;2548 ; FUNCTIONAL DESCRIPTION:
            ;2549 ;
            ;2550 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2551 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2552 ; and finally HOLD for the last 16 cycles.
            ;2553 ;
            ;2554 ; CALLING CONSTRAINT:
            ;2555 ;
            ;2556 ; =0
            ;2557 ;
            ;2558 ; SIDE EFFECTS:
            ;2559 ;
            ;2560 ; D Reg destroyed
            ;2561 ;--
            ;2562   ;
            ;2563   ; assert hold for 16 cycles
            ;2564   ;
            ;2565 SBI.UNJAM:
            ;2566  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11F4, 0818,0038,6580,8000,0010,10AE  ;2567      clk.ubcc  ; set micro cc's for next cycle
            ;2568 =0
            ;2569 SBI.UNJAM.1:
            ;2570  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2571       clk.ubcc,z?, ; ...
U 10AE, 0819,0100,0580,8000,0010,10AE  ;2572      j/sbi.unjam.1 ; test for completion
            ;2573    ;
            ;2574    ; assert hold and unjam for 16 cycles
            ;2575    ;
            ;2576  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10AF, 0818,0038,6580,8800,0010,10B0  ;2577      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2578 =0
            ;2579 SBI.UNJAM.2:
            ;2580  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2581       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10B0, 0819,0100,0580,8800,0010,10B0  ;2582      z?,j/sbi.unjam.2 ; ...
            ;2583    ;
            ;2584    ; assert hold for 16 cycles
            ;2585    ;
            ;2586  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10B1, 0818,0038,6580,8000,0010,10B2  ;2587      clk.ubcc  ; and set cc's for next cycle
            ;2588 =0
            ;2589 SBI.UNJAM.3:
            ;2590  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2591       clk.ubcc,z?, ; ...
U 10B2, 0819,0100,0580,8000,0010,10B2  ;2592      j/sbi.unjam.3 ; ...
U 10B3, 0000,003E,0180,0800,0000,0001  ;2593  return1   ; exit
```

```
;2594  .PAGE
;2595  .TOC "   RESET SUBTEST NUMBER"
;2596 ;++
;2597 ; FUNCTIONAL DESCRIPTION:
;2598 ;
;2599 ; Since some  of the tests will have to be restarted when two
;2600 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
;2601 ; the subtest counter to zero ( only for thoes tests that have
;2602 ; more than one subtest). This subroutine may also be necessary
;2603 ; when a test is being loop on.
;2604 ;
;2605 ; CALLING CONSTRAINT:
;2606 ;
;2607 ; =0
;2608 ; SIDE EFFECTS:
;2609 ;
;2610 ; D is destroyed
;2611 ;
;2612 ;--
;2613 RESET.SUBTST:
;2614  RC[0F]_0,   ; Reset subtest number
U 11F5, 0003,003E,0180,09F8,0000,0001  ;2615   RETURN1   ; ...and return
```

```
            ;2616  .PAGE
            ;2617  .TOC "   Initialize the SBI"
            ;2618 ;++
            ;2619 ; FUNCTIONAL DESCRIPTION:
            ;2620 ;
            ;2621 ; This routine performs the following functions:
            ;2622 ;
            ;2623 ;  Executes an SBI Unjam
            ;2624 ;  Clears the SBI Fault Latch
            ;2625 ;  Clears the SBI Error Register
            ;2626 ;
            ;2627 ; CALLING CONSTRAINT:
            ;2628 ;
            ;2629 ; =0
            ;2630 ;
            ;2631 ; SIDE EFFECTS:
            ;2632 ;
            ;2633 ; D & Q Register destroyed
            ;2634 ;--
            ;2635 =0
            ;2636 SBI.INIT:
U 10B4. 0000.003D.0180.0800.0000.11F4  ;2637  call[sbi.unjam]  ; Unjam the SBI
U 10B5. 0818.0038.C180.0800.0000.11F6  ;2638  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11F6. 0801.0028.6580.3C00.0000.11F7  ;2639  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11F7. 0F00.003C.6D80.3C00.0000.11F8  ;2640  id[fault]_d,d_0
U 11F8. 0000.003C.6D80.3C00.0000.11F9  ;2641  id[fault]_d
U 11F9. 0000.003E.6580.3C00.0000.0001  ;2642  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2643  .PAGE
            ;2644  .TOC "   Disable Cache Routine"
            ;2645 ;++
            ;2646 ; FUNCTIONAL DESCRIPTION:
            ;2647 ;
            ;2648 ; This routine disables cache hits by setting bits <16:15>
            ;2649 ; in the maintenance register.
            ;2650 ;
            ;2651 ; CALLING SEQUENCE:
            ;2652 ;
            ;2653 ; =0
            ;2654 ;
            ;2655 ; SIDE EFFECTS:
            ;2656 ;
            ;2657 ; D & Q Registers destroyed
            ;2658 ;--
            ;2659 DISABLE.CACHE:
U 11FA. 0818.0038,4580.0800,0000.11FB  ;2660  d_k[.8000]  ; ... and seed constant
U 11FB. 0500.003C,0180.0800,0000.11FC  ;2661  d_d.left  ; Generate final constant
U 11FC. 0819.0030,4580.0800,0000.11FD  ;2662  d_d.or.k[.8000]  ; ... = 00018000
U 11FD. 0000.003E,7580.3C00,0000.0001  ;2663  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2664  .PAGE
            ;2665  .TOC "   Enable Cache"
            ;2666 ;++
            ;2667 ; FUNCTIONAL DESCRIPTION:
            ;2668 ;
            ;2669 ; This routine enables cache group 0 by clearing the force
            ;2670 ; miss bit in the maintenance register.
            ;2671 ;
            ;2672 ; CALLING CONSTRAINT:
            ;2673 ;
            ;2674 ; =0
            ;2675 ;
            ;2676 ; SIDE EFFECTS:
            ;2677 ;
            ;2678 ; D, Q AND SC Registers destroyed
            ;2679 ;--
            ;2680 ENABLE.CACHE:
U 11FE, 0000.003C.75F0.2C00.0000.11FF  ;2681  q_id[maint]  ; Get current maintenance register
U 11FF, 0000.003C.6580.0800.0084.7200  ;2682  sc_k[.10]  ; Setup to clear bit 16
U 1200, 0801.2034.0180.0800.0000.1201  ;2683  d_q.and.mask  ; Clear bit 16
U 1201, 0000.003E.7580.3C00.0000.0001  ;2684  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2685  .PAGE
            ;2686  .TOC "   Wait Loop Routine"
            ;2687 ;++
            ;2688 ; FUNCTIONAL DESCRIPTION:
            ;2689 ;
            ;2690 ; As the name implies,  this subroutine is used to set up a wait
            ;2691 ; loop for a specified  number of cycles.  This may be necessary
            ;2692 ; when the micro-program has to wait for another event to finish.
            ;2693 ; When the subroutine is entered Register D should be holding the
            ;2694 ; desired numbered of cycles.
            ;2695 ;
            ;2696 ; CALLING CONSTRAINT:
            ;2697 ;
            ;2698 ; =0
            ;2699 ;
            ;2700 ; INPUTS:
            ;2701 ;
            ;2702 ; d - Contains number of cycles to wait
            ;2703 ; alu.z condition code must not be set
            ;2704 ;
            ;2705 ; SIDE EFFECTS:
            ;2706 ;
            ;2707 ; d is destroyed
            ;2708 ;--
            ;2709 WAIT.LOOP:
U 1202, 0018,0038,6180,0800,0010,10B6  ;2710  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2711     ; ...is not set
            ;2712 =0
            ;2713 W.LOOP:
            ;2714  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10B6, 0819,0100,0580,0800,0010,10B6  ;2715       j/W.LOOP
U 10B7, 0000,003E,0180,0800,0000,0001  ;2716  return1   ; exit
            ;2717
```

```
            ;2718  .PAGE
            ;2719  .TOC "   Check for Interrupt Routine"
            ;2720 ;**
            ;2721 ; FUNCTIONAL DESCRIPTION:
            ;2722 ;
            ;2723 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2724 ; If an interrupt is active, the following registers are setup:
            ;2725 ;  RC[8] - Gets the VECTOR register
            ;2726 ;  RC[7] - Gets the SBI ERROR register
            ;2727 ;  RC[6] - Gets the FAULT register
            ;2728 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2729 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2730 ;
            ;2731 ; CALLING CONSTRAINT:
            ;2732 ;
            ;2733 ; =00
            ;2734 ;
            ;2735 ; SIDE EFFECTS:
            ;2736 ;
            ;2737 ; D & Q are destroyed
            ;2738 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2739 ;--
            ;2740 CHECK_INTERRUPT:
            ;2741    ;
            ;2742    ; First, enable the fault and RDS/CRD interrupts
            ;2743    ;
U 1203, 0818,0038,4580,0800,0000,1204  ;2744  d_k[.8000]  ; Get data to set interrupt enable
            ;2745  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1204, 0500,003C,6580,3C00,0000,1205  ;2746       d_d.left
U 1205, 0100,003C,0180,0800,0000,1206  ;2747  d_d.left2  ; D = 40000
U 1206, 0000,003C,6D80,3C00,0000,1207  ;2748  id[fault]_d  ; Set fault interrupt enable
            ;2749  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1207, 0F03,003C,0180,09A8,4000,1208  ;2750       rc[5]_0  ; and clear interrupt occurred flag
U 1208, 0000,003C,3D80,3C00,0000,1209  ;2751  id[psl]_d  ; Clear the PSL
U 1209, 0000,003C,6580,3C00,0000,120A  ;2752  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 120A, 0000,003C,6D80,3C00,0000,120B  ;2753  id[fault]_d  ; Clear FAULT Interrupt Enable
U 120B, 0000,0E3C,0180,0800,0000,1004  ;2754  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2755 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2756  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2757  return1   ; No interrupt
            ;2758 =111
            ;2759 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,120C  ;2760  q_id[vector]  ; Get ID Vector Register
U 120C, 0001,203C,65F0,2DC0,0000,120D  ;2761  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 120D, 0001,203C,6DF0,2DB8,0000,120E  ;2762  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 120E, 0001,203C,0180,09B0,0000,120F  ;2763  rc[6]_q   ; Save falut reg
U 120F, 0018,003A,0580,09A8,0000,0002  ;2764  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2765  .PAGE
            ;2766  .TOC "   CHECK UTRAP"
            ;2767 ;**
            ;2768 ;FUNCTIONAL DESCRIPTION:
            ;2769 ;
            ;2770 ; This subroutine is used to check wether a Utrap occured.
            ;2771 ; It takes the contents of the Save Utrap flags register
            ;2772 ; R[0E], and compares them to the contrnts of the Utrap
            ;2773 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2774 ; according to the results of the test  (i.e., contents of
            ;2775 ; these registers are equal or not).
            ;2776 ; CALLING CONSTRAINT:
            ;2777 ;
            ;2778 ; =00
            ;2779 ;
            ;2780 ; INPUTS:
            ;2781 ;
            ;2782 ; R[0E]- Saved Utrap Mask
            ;2783 ; R[0F]- Expected Utrap Mask
            ;2784 ;
            ;2785 ;--
            ;2786 CHECK_UTRAP:
U 1210,  0800,003C,0180,0A70,0000,1211  ;2787  D_R[0E]    ; Reg D is loaded with the trap mask
U 1211,  0001,003C,0180,09C0,0000,1212  ;2788  RC[08]_D   ; Save expected Utrap flag for error logout
U 1212,  0800,003C,0180,0A78,0000,1213  ;2789  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1213,  0001,003C,0180,09B8,0000,1214  ;2790  RC[07]_D   ; Save in RC[07]
            ;2791  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1214,  000D,0020,0180,0A70,0010,1215  ;2792         CLK.UBCC
U 1215,  0003,013C,0180,0AF8,0000,10B8  ;2793  Z?,R[0F]_0   ; Branch if no traps
U 10B8,  0000,003E,0180,0800,0000,0002  ;2794 =0 RETURN2    ; Utrap occured
U 10B9,  0000,003E,0180,0800,0000,0001  ;2795  RETURN1    ; No Utrap  return
            ;2796
```

```
            ;2797  .PAGE
            ;2798  .TOC "   Set Trap Mask"
            ;2799 ;**
            ;2800 ; FUNCTIONAL DESCRIPTION:
            ;2801 ;
            ;2802 ; This routine is used to set a bit in the Micro Trap Mask
            ;2803 ; R[0F].
            ;2804 ;
            ;2805 ; CALLING CONSTRAINT:
            ;2806 ;
            ;2807 ; =0
            ;2808 ;
            ;2809 ; INPUTS:
            ;2810 ;
            ;2811 ; SC - Contains bit number to set.
            ;2812 ;
            ;2813 ; OUTPUTS:
            ;2814 ;
            ;2815 ; rc[0E] - Contains the current trap mask
            ;2816 ;
            ;2817 ; SIDE EFFECTS:
            ;2818 ;
            ;2819 ; d regsiter is destroyed
            ;2820 ;--
            ;2821 SET_TRAP_MASK:
U 1216, 0800,003C,0180,0A78,0000,1217  ;2822  d_r[0f]
U 1217, 0801,001C,0180,0AF8,0000,1218  ;2823  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1218, 0001,003E,0180,0AF0,0000,0001  ;2824  r[0E]_d,return1    ; Save mask and return
```

```
            ;2825  .PAGE
            ;2826  .TOC "   FIND MS780-E MEMORY"
            ;2827 ;**
            ;2828 ; FUNCTIONAL DESCRIPTION:
            ;2829 ;
            ;2830 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2831 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2832 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2833 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2834 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2835 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2836 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2837 ;     (zero).
            ;2838 ;     All the other MS780-E/H's found will have their starting  address
            ;2839 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2840 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2841 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2842 ;     current MS780-E unit under test.
            ;2843 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2844 ;     set to maximum.
            ;2845 ;
            ;2846 ; CALLING CONSTRAINT:
            ;2847 ;
            ;2848 ; =00
            ;2849 ;
            ;2850 ; OUTPUTS:
            ;2851 ;
            ;2852 ; rc[0e] - Contains address of Configuration Register
            ;2853 ; r[0]  - Contains TR level
            ;2854 ;
            ;2855 ; SIDE EFFECTS:
            ;2856 ;
            ;2857 ; D register is destroyed.
            ;2858 ;--
            ;2859 =0
            ;2860 FIND.MS780E:
U 10BA, 0000.003D.0180.0800.0000.10B4  ;2861  call[sbi.init]  ; Make sure SBI is enabled
U 10BB, 0003.003C.0180.0988.0000.1219  ;2862  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1219, 0818.0038.1980.0800.0000.121A  ;2863  d_k[zero]  ; Clear MS780-E/H counter
U 121A, 0000.003C.F980.3C00.0000.121B  ;2864  id[3e]_d  ; ...
U 121B, 0018.0038.0580.0A80.0000.121C  ;2865  r[0]_k[.1]  ; Init TR counter
U 121C, 0F03.003C.0180.09F0.0000.10BC  ;2866  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2867     ; ...CNFG A Reg
            ;2868 =0
            ;2869 FIND.MEM.LOOP:
U 10BC, 0800.003D.0180.0A00.0000.1242  ;2870  d_r[0],call[generate.io]; Generate address of TR level
U 10BD, 0001.003C.0180.09F0.0200.10BE  ;2871  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10BE, 0000.003D.8980.0800.0084.7216  ;2872 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2873  d[long]_cache.p, ; Read the specified tr level
U 10BF, 0000.003C.0180.C970.0000.121D  ;2874   ic_rc[0e]  ; ...
U 121D, 0000.003C.0180.0A78.0010.121E  ;2875  alu_r[0f],clk.ubcc ; Check for no response
U 121E, 0001.013C.0180.0880.0082.10C0  ;2876  z?,sc_d.ia_ra[0] ; Branch if no adapter here
U 10C0, 0000.003C.0180.0800.0000.121F  ;2877 =0 j/check.adpt.code ; Check for a memory
U 10C1, 0000.003C.0180.0800.0000.123E  ;2878  j/find.mem.lp1  ; Try next tr
            ;2879 CHECK.ADPT.CODE:
```

```
U 121F. 0000.003C.1D80.0800.1404.7220  ;2880  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1220. 0000.163C.0180.0800.0000.1018  ;2881  state7-4?  ; Branch on the adapter type
U 1018. 0000.003C.8980.0800.0084.7221  ;2882 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019. 0000.003C.8980.0800.0084.7222  ;2883  j/ms780.c,sc_k[.d] ; MS780-C
U 101A. 0000.003C.0180.0800.0000.123E  ;2884  j/find.mem.lp1  ; Not a memory
U 101B. 0000.003C.0180.0800.0000.123E  ;2885  j/find.mem.lp1  ; Not a memory
U 101C. 0000.003C.6580.0800.0084.7227  ;2886  j/ma780.max,sc_k[.10] ; MA780
U 101D. 0000.003C.0180.0800.0000.123E  ;2887  j/find.mem.lp1  ; Not a memory
U 101E. 0010.0038.0180.09F0.0000.122B  ;2888  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F. 0010.0038.0180.09F0.0000.122B  ;2889  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2890   ;
          ;2891   ; Found an MS780-A or -C. Set the starting address
          ;2892   ; to maximum.
          ;2893   ;
U 1221. 0818.0038.5D80.0800.0000.1223  ;2894 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1222. 0818.0038.0580.0800.0000.1223  ;2895 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
          ;2896 MS780.COMMON:
U 1223. 0819.0030.7180.0800.0000.1224  ;2897  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1224. 0000.003C.01F8.0803.0080.D225  ;2898  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1225. 0D00.003C.0180.0800.0000.1226  ;2899  d_dal.sc  ; Put data in bits<29:14>
          ;2900  cache.p_d[long], ; Set the starting address to maximum
U 1226. 0000.003C.0180.A800.0000.123E  ;2901       j/find.mem.lp1 ; and continue TR scan
          ;2902   ;
          ;2903   ; Found an MA780. Set the starting address to maximum
          ;2904   ;
          ;2905 MA780.MAX:
U 1227. 0818.0038.39F8.0803.0000.1228  ;2906  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1228. 0D00.003C.0180.0803.0000.1229  ;2907  d_dal.sc,va_va+4 ; Put in proper position
U 1229. 0000.003C.0180.0803.0000.122A  ;2908  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;2909  cache.p_d[long], ; Set starting address to maximum
U 122A. 0000.003C.0180.A800.0000.123E  ;2910       j/find.mem.lp1
          ;2911 ;
          ;2912 ; Found an MS780E
          ;2913 ;
          ;2914 FOUND.MS780E:
U 122B. 0000.003C.F9F0.2C00.0000.122C  ;2915  q_id[3e]  ; Set up to see how many MS780-E's
          ;2916  alu_q.xor.k[.3], ; Are there Maximum units?
U 122C. 0019.2020.0D80.0800.0010.122D  ;2917   clk.ubcc ; Check condition codes
U 122D. 0000.013C.0180.0800.0000.10C2  ;2918  z?   ; Are we at maximum units for system
U 10C2. 0000.003C.0180.0800.0000.122E  ;2919 =0 J/ms780e.tst  ; No go find how many units ther are
U 10C3. 0818.0038.C180.0800.0000.10C4  ;2920  d_k[.ffff]  ; Yes set address to maximum
U 10C4. 0000.003D.0180.0800.0000.1246  ;2921 =0 call[set.start.addr] ; .....
U 10C5. 0000.003C.0180.0800.0000.123E  ;2922  j/find.mem.lp1  ; Go check to see if we are done
          ;2923 ;
          ;2924 MS780E.TST:
          ;2925  alu_rc[01],  ; Check "Found MS780E Flag"
U 122E. 0010.0038.0180.0908.0010.122F  ;2926   clk.ubcc ; Check condition codes
U 122F. 0810.0138.0180.0970.0000.10C6  ;2927  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;2928 =0 j/not.first.ms780e, ; No
U 10C6. 0818.0038.C180.0800.0000.10CA  ;2929   d_k[.ffff] ; Set starting address
U 10C7. 0001.003C.0180.0988.0000.1230  ;2930  rc[01]_d  ; Yes put base address in rc[01]
U 1230. 0818.0038.7980.0800.0000.1231  ;2931  d_k[.30]  ; Set up SC to point to first unit
U 1231. 0019.0014.F580.0800.0082.1232  ;2932  alu_d+k[.a],sc_alu ; ...
U 1232. 0810.0038.0180.0908.0000.1233  ;2933  d_rc[01]  ; Put base address of unit in D
U 1233. 0000.003C.0180.3400.0000.1234  ;2934  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1234, 0F00.003C.0180.0800.0000.10C8  ;2935  d_0   ; Prepare to set starting address to Zero
U 10C8, 0000.003D.0180.0800.0000.1246  ;2936 =0 call[set.start.addr] ; Go do it
        ;2937  j/find.mem.lp1,  ; Check to see if we are done
U 10C9, 0003.003C.0180.09E8.0000.123E  ;2938   rc[0d]_0 ; ....
        ;2939 =0
        ;2940 NOT.FIRST.MS780E:
U 10CA, 0000.003D.0180.0800.0000.1246  ;2941  call[set.start.addr] ; Go set starting address to maximum
U 10CB, 0000.003C.F9F0.2C00.0000.1235  ;2942  q_id[3e]  ; Get the number of MS780-Es found
U 1235, 0819.2014.0580.0800.0000.1236  ;2943  d_q+k[.1]  ; Increment this counter
U 1236, 0000.003C.F980.3C00.0000.1237  ;2944  id[3e]_d  ; Save the counter
U 1237, 0018.0038.11C0.0800.0000.1238  ;2945  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2946      ; ...were the I/O base addresses will be stored
U 1238, 081D.2000.7980.0800.0000.1239  ;2947  d_q-d,k[.30]  ; ... adjust pointer
U 1239, 0819.0014.7980.0800.0000.123A  ;2948  d_d+k[.30]  ; Point the SC to the ID register
U 123A, 0000.003C.F580.0800.0000.123B  ;2949  k[.a]   ; ...to receive I/O base address
U 123B, 0019.0014.F580.0800.0082.123C  ;2950  alu_d+k[.a],sc_alu ; ...
U 123C, 0810.0038.0180.0970.0000.123D  ;2951  d_rc[0e]  ; Get base address to d
U 123D, 0000.003C.0180.3400.0000.123E  ;2952  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2953   ;
        ;2954   ; Try next tr
        ;2955   ;
        ;2956 FIND.MEM.LP1:
U 123E, 0818.0014.0580.0A80.0000.123F  ;2957  r[0]_ia+k[.1],d_alu ; Increment TR level
        ;2958  alu_d.and.k[.f],
U 123F, 0019.0034.6180.0800.0010.1240  ;2959      clk.ubcc  ; Check if all done
U 1240, 0000.013C.0180.0800.0000.10CC  ;2960  z?   ; Branch if yes
U 10CC, 0000.003C.0180.0800.0000.10BC  ;2961 =0 j/find.mem.loop  ; Try next TR
U 10CD, 0810.0038.0180.0908.0010.1241  ;2962  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1241, 0000.013C.0180.0800.0000.10CE  ;2963  z?   ; Check condition codes
U 10CE, 0001.003E.0180.09F0.0000.0002  ;2964 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10CF, 0000.003E.0180.0800.0000.0001  ;2965  return1   ; No MS780E found
        ;2966
```

```
            ;2967  .PAGE
            ;2968  .TOC "   Generate IO Address"
            ;2969 ;++
            ;2970 ; FUNCTIONAL DESCRIPTION:
            ;2971 ;
            ;2972 ; This routine is used to generate an SBI Configuration Register
            ;2973 ; address from a TR level.
            ;2974 ;
            ;2975 ; CALLING CONSTRAINT:
            ;2976 ;
            ;2977 ; =0
            ;2978 ;
            ;2979 ; INPUTS:
            ;2980 ;
            ;2981 ; D - Contains TR level
            ;2982 ;
            ;2983 ; OUTPUTS:
            ;2984 ;
            ;2985 ; D - Contains SBI IO address
            ;2986 ;--
            ;2987 GENERATE.IO:
U 1242, 0000,003C,89F8,0800,0084,7243  ;2988  sc_k[.d],q_0   ; Setup to shift TR level 13 bits
U 1243, 0D00,003C,8D80,0800,0084,7244  ;2989  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 1244, 0000,003C,0980,0800,0084,B245  ;2990  sc_sc-k[.2]  ; insert bit 29
U 1245, 0801,001E,0180,0800,0000,0001  ;2991  d_d.ornot.mask,return1 ; and return
```

```
            ;2992  .PAGE
            ;2993  .TOC "   Set Starting Address"
            ;2994 ;++
            ;2995 ; FUNCTIONAL DESCRIPTION:
            ;2996 ;
            ;2997 ; This routine is used to set the starting address of the
            ;2998 ; memory to the specified value.
            ;2999 ;
            ;3000 ; CALLING CONSTRAINT:
            ;3001 ;
            ;3002 ; =0
            ;3003 ;
            ;3004 ; INPUTS:
            ;3005 ;
            ;3006 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3007 ;     (B Register Image divided by 2**16)
            ;3008 ; rc[0e] - Contains Address of Register A
            ;3009 ;
            ;3010 ; OUTPUTS:
            ;3011 ;
            ;3012 ; d - Contains aligned starting address (B Register Image)
            ;3013 ;
            ;3014 ; SIDE EFFECTS:
            ;3015 ;
            ;3016 ; d,q,va, and sc are destroyed
            ;3017 ;--
            ;3018 SET.START.ADDR:
U 1246, 0010,0038,6580,0970,0284,7247  ;3019  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1247, 0000,003C,01F8,0803,0000,1248  ;3020  va_va+4,q_0  ; Set address to Register B
            ;3021  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1248, 0D00,003C,0980,0800,0084,B249  ;3022      sc_sc-k[.2]  ; Setup to insert bit 14
U 1249, 0801,001C,0180,0800,0000,124A  ;3023  d_d.ornot.mask  ; Insert Write Enable bit
U 124A, 0000,003E,0180,A800,0000,0001  ;3024  cache.p_d[long],return1 ; Set the starting address and return
```

```
            ;3025  .PAGE
            ;3026  .TOC "   ENABLE NEXT MS780-E"
            ;3027 ;**
            ;3028 ; FUNCTIONAL DESCRIPTION:
            ;3029 ;
            ;3030 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3031 ;    They will be tested  separately, according to the TR level at which
            ;3032 ;    they are located,  starting with the one  at the lowest level first.
            ;3033 ;          When  a test has finished  with the first unit, it will have to call
            ;3034 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3035 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3036 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3037 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3038 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3039 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3040 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3041 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3042 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3043 ;
            ;3044 ; CALLING CONSTRAINT:
            ;3045 ;
            ;3046 ;  =00
            ;3047 ;
            ;3048 ;--
            ;3049 ENABLE.NEXT.MS780E:
U 124B, 0000,003C,F9F0,2C00,0000,124C  ;3050  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3051  ALU_Q,   ; Check to see if it is zero
U 124C, 0001,203C,0180,0800,0010,124D  ;3052     CLK.UBCC  ; Check Condition Codes
U 124D, 0000,013C,0180,0800,0000,10D0  ;3053  Z?   ; ...for zero
U 10D0, 0000,003C,0180,0800,0000,124E  ;3054 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10D1, 0000,003E,0180,0800,0000,0002  ;3055  RETURN2   ; Zero No more units to test
            ;3056 ENABLE.NEXT:
U 124E, 0818,0038,C180,0800,0000,10D2  ;3057  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10D2, 0000,003D,0180,0800,0000,1246  ;3058 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10D3, 0818,0038,7980,0800,0000,124F  ;3059  D_K[.30]  ; Prepare to point to the ID register holding
            ;3060     ; ...the I/O Base address of the next MS780-E
U 124F, 0819,0014,1180,0800,0000,1250  ;3061  D_D+K[.4]  ; ...
U 1250, 0000,003C,F580,0800,0000,1251  ;3062  K[.A]   ; ...
U 1251, 0819,0014,F580,0800,0000,1252  ;3063  D_D+K[.A]  ; ...
U 1252, 0000,003C,F9F0,2C00,0000,1253  ;3064  Q_ID[3E]  ; Get MS780-E Counter
            ;3065  D_D-Q,   ; D now holds the pointer to the ID register
U 1253, 081D,0000,0180,0800,0082,1254  ;3066    SC_ALU  ; ...
U 1254, 0000,003C,01F0,2400,0000,1255  ;3067  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1255, 0001,203C,0180,09F0,0000,1256  ;3068  RC[0E]_Q  ; Save it also in RC[0E]
U 1256, 0F03,003C,0180,09E8,0000,10D4  ;3069  RC[0D]_0,D_0  ; Set starting address to zero
U 10D4, 0000,003D,0180,0800,0000,1246  ;3070 =0 CALL[SET.START.ADDR] ; ....
U 10D5, 0F00,003C,F9F0,2C00,0000,1257  ;3071  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1257, 081D,2008,0180,0800,0000,1258  ;3072  D_Q-D-1   ; Subtract 1 from total
U 1258, 0000,003C,F980,3C00,0000,10D6  ;3073  ID[3E]_D  ; Adjust the pointer
U 10D6, 0000,003D,0180,0800,0000,11F5  ;3074 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10D7, 0000,003E,0180,0800,0000,0001  ;3075  RETURN1   ; Tell caller another unit was found
            ;3076
```

```
;3077  .PAGE
;3078  .TOC "   MS780-E/H CNFG REG CLEANUP"
;3079 ;*************************************************************
;3080 ;**
;3081 ;
;3082 ; Functional Description:
;3083 ; ----------------------
;3084 ;
;3085 ;      This subroutine is used to clear all error conditions
;3086 ; in the MS780-E/H Configuration/Status/Error Registers.
;3087 ; Bits beeing cleared are:
;3088 ;
;3089 ; CNFG Regsiter B <11:00>
;3090 ;
;3091 ; CNFG Register C and D <31:28> and <10:06>
;3092 ;   ---
;3093 ;
;3094 ;
;3095 ; Calling Constraint: =0
;3096 ; ------------------
;3097 ;
;3098 ;
;3099 ; Inputs:
;3100 ; ------
;3101 ;
;3102 ; RC[0E] MUST contain the I/O base address of
;3103 ; the MS780-E/H currently under test
;3104 ;
;3105 ; Outputs:
;3106 ; -------
;3107 ;
;3108 ; NO specific outputs.
;3109 ; Above mentioned bits will be cleared.
;3110 ;
;3111 ; Side Effects:
;3112 ; ------------
;3113 ;
;3114 ; The contents of the following registers will be lost
;3115 ; Q, D, SC, VA
;3116 ;
;3117 ;
;3118 ;--
;3119 ;*************************************************************
;3120 MS780E.CLEANUP:
U 1259, 0010,0038,0180,0970,0200,125A  ;3121  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
          ;3122  D_0,     ; Get data to clear Read/Write bits
U 125A, 0F00,003C,0180,0803,0000,125B  ;3123   VA_VA+4    ; Point to CNFG Reg B
U 125B, 0000,003C,0180,A800,0000,125C  ;3124  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 125C, 0818,0038,7980,0800,0000,125D  ;3125  D_K[.30]    ; Prepare to form x30000780 to clear
          ;3126       ; ...CNFG Reg's C and D
U 125D, 0B00,003C,0180,0800,0000,125E  ;3127  D_D.SWAP    ; D = x30000000
          ;3128  Q_D,     ; Save in the Q Reg
U 125E, 0818,0038,CDE0,0800,0000,125F  ;3129   D_K[.F0]  ; D = xF0
U 125F, 0000,003C,0D80,0800,0084,7260  ;3130  SC_K[.3]    ; Must shift D Reg left 3 bits to get
          ;3131       ; ...x780
```

```
U 1260, 0D00,003C,0180,0800,0000,1261  ;3132  D_DAL.SC    ; D = x780
        ;3133  D_D.OR.Q,    ; D = x30000780
U 1261, 081D,0030,0180,0803,0000,1262  ;3134   VA_VA+4    ; Point to CNFG Reg C
U 1262, 0000,003C,0180,A800,0000,1263  ;3135  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 1263, 0000,003C,0180,0800,0000,1264  ;3136  NOP
U 1264, 0000,003C,0180,A800,0000,1265  ;3137  CACHE.P_D[LONG]   ; Do second write to make sure that
        ;3138       ; ...uSequencer Parity error bit
        ;3139       ; ...is clear
U 1265, 0000,003C,0180,0803,0000,1266  ;3140  VA_VA+4    ; Point to CNFG Reg D
U 1266, 0000,003C,0180,A800,0000,1267  ;3141  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 1267, 0000,003C,0180,0800,0000,1268  ;3142  NOP
U 1268, 0000,003C,0180,A800,0000,1269  ;3143  CACHE.P_D[LONG]   ; Do second write to clear uSequencer
        ;3144       ; ...Parity error bit
U 1269, 0000,003E,0180,0800,0000,0001  ;3145  RETURN1    ; Return to caller
        ;3146
```

```
;3147  .PAGE "TEST 1A5 MICRO SEQUENCER PARITY ERROR"
;3148 ;*************************************************************
;3149 ;***
;3150 ;
;3151 ; Functional Description:
;3152 ; ----------------------
;3153 ;     The test checks the operation of the Micro-sequencer Parity
;3154 ;    error bit (bit 07 in Cnfg C or D) by forcing a micro-sequencer
;3155 ;  parity error (set bit 31 of Cnfg C or D).
;3156 ;
;3157 ;
;3158 ; Logic Description:
;3159 ; -----------------
;3160 ;
;3161 ;  N/A
;3162 ;
;3163 ; Error Logout:
;3164 ; ------------
;3165 ;
;3166 ; See individual error reports.
;3167 ;
;3168 ; Sync Point Description:
;3169 ; ----------------------
;3170 ;
;3171 ;     N/A
;3172 ;
;3173 ;=============================================================
;3174 ;
;3175 ; Register Map:
;3176 ; ------------
;3177 ;
;3178 ;
;3179 ;       RC       Description:
;3180 ;       --       -----------
;3181 ;       E Base address of currently tested MS780-E
;3182 ;       D CNFG Register C/D address
;3183 ;       C Expected Masked Register contents
;3184 ;       B  Recieved Masked Register contents
;3185 ;       A Actual Recieved Register contents
;3186 ;       5 Mask for bit 31 for CNFG C/D
;3187 ;       6 Mask for bit 20 for CNFG A
;3188 ;       7 Mask for bit 31 and 7 for CNFG C/D
;3189 ;--
;3190 ;*************************************************************
;3191 =0
U 10D8, 0000,003D,0180,0800,0000,109B  ;3192 TST.18: NEWTST   ;
U 10D9, 0000,003C,0180,0800,0000,102C  ;3193  NOP
;3194 =00
U 102C, 0000,003D,0180,0800,0000,10BA  ;3195     CALL[FIND.MS780E] ; See if there are any MS780-E's
;3196      ; ...on the SBI
U 102D, 0000,003C,0180,0800,0000,1290  ;3197  J/TEST.18.EXIT  ; No MS780-E's on the SBI
U 102E, 0818,0038,4180,0800,0000,102F  ;3198  D_K[.80]  ; Begin to set up Test Constants
U 102F, 0B00,003C,0180,0800,0000,126A  ;3199  D_D.SWAP  ; ...
U 126A, 0001,003C,0180,09A8,0000,126B  ;3200  RC[05]_D  ; Bit 31 for Cnfg Reg C/D
U 126B, 0819,0030,4180,0800,0000,126C  ;3201  D_D.OR.K[.80]  ; ...
```

```
U 126C. 0001.003C.2180.09B8.0084.726D  ;3202  RC[07]_D,SC_K[.14] ; Bit 31 and 7 in CNFG Reg C/D
U 126D. 0003.001C.0180.09B0.0000.1030  ;3203  RC[06]_NOT.MASK  ; Bit 20 for Cnfg Reg A
        ;3204 =00
        ;3205 RESTART.TEST.18:  ; Entry point for second controller
        ;3206      ; ... or next Ms780-E
U 1030. 0000.003D.0180.0800.0000.10AA  ;3207  CALL[START.TEST] ; Configure Controllers
U 1031. 0000.003C.0180.0800.0000.1290  ;3208  J/TEST.18.EXIT  ; No more controllers or MS780-E's
U 1032. 0000.003C.0180.0800.0000.10DA  ;3209  NOP
        ;3210 =
U 10DA. 0000.003D.0180.0800.0000.11D3  ;3211 =0 ERLOOP   ; Pass Uaddress for loop on error
U 10DB. 0000.003C.0180.0800.0000.10DC  ;3212  NOP
U 10DC. 0000.003D.0180.0800.0000.1259  ;3213 =0 CALL[MS780E.CLEANUP] ; Clear memory CNFG Regs
U 10DD. 0010.0038.0180.0968.0200.126E  ;3214  VA_RC[0D]  ; Point to CNFG Reg C/D
U 126E. 0810.0038.0180.0928.0000.126F  ;3215  D_RC[05]  ; Get bit to set in CNFG C/D Reg
U 126F. 0000.003C.0180.A800.0000.1270  ;3216  CACHE.P_D[LONG]  ; Set "Force Usequencer Parity" bit
U 1270. 0003.003C.0180.09C8.0000.1271  ;3217  RC[9]_0   ; Clear flag for the Fail Chain
U 1271. 0018.0038.1980.0800.0200.1272  ;3218  VA_K[ZERO]  ; ....
U 1272. 0000.003C.0180.A800.0000.1273  ;3219  CACHE.P_D[LONG]  ; Write location and force the error
U 1273. 0010.0038.0180.0968.0200.1274  ;3220  VA_RC[0D]  ; Point to Cnfg Reg C/D
U 1274. 0000.003C.0180.C800.0000.1275  ;3221  D[LONG]_CACHE.P  ; Read the Data
U 1275. 0001.003C.0180.09D0.0000.1276  ;3222  RC[0A]_D  ; Save the actual register data
U 1276. 0010.0038.01C0.0938.0000.1277  ;3223  Q_RC[07]  ; Get the mask
U 1277. 001D.0034.0180.09D8.0000.1278  ;3224  RC[0B]_D.AND.Q        ; Mask the unwanted bits
        ;3225  ALU_Q[XOR]RC[0B],        ; Check Bits 31 and 7 should be set
U 1278. 0011.2020.0180.0958.0010.1279  ;3226   CLK.UBCC ; Clock condition codes
U 1279. 0001.213C.0180.09E0.0000.10DE  ;3227  Z?,RC[0C]_Q  ; Branch if bits are set
```

```
            ;3228  .PAGE
            ;3229 ;
            ;3230 ;////////////////////////////////////////////////////////////////////////////////
            ;3231 ;
U 10DE, 0000,003D,D580,0800,0084,70A6  ;3232 =0 ERROR2[.6]  ; Error bits in register C incorrect
            ;3233 ;
            ;3234 ;
            ;3235 ;////////////////////////////////////////////////////////////////////////////////
            ;3236 ;
            ;3237 ; ERROR LOGOUT:
            ;3238 ;
            ;3239 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;3240 ;  CNFG REG C/D ADDRESS
            ;3241 ;  EXPECTED MASKED REGISTER DATA
            ;3242 ;  RECIEVED MASKED REGISTER DATA
            ;3243 ;  ACTUAL REGIASTER DATA
            ;3244 ;  NOT USED
            ;3245 ;
            ;3246 ;////////////////////////////////////////////////////////////////////////////////
            ;3247 ;
U 10DF, 0010,0038,0180,0970,0200,127A  ;3248  VA_RC[0E]  ; Point to Cnfg Reg A
U 127A, 0000,003C,0180,C800,0000,127B  ;3249  D[LONG]_CACHE.P  ; Read the register
U 127B, 0001,003C,0180,09D0,0000,127C  ;3250  RC[0A]_D  ; Get actual register Data
U 127C, 0010,0038,01C0,0930,0000,127D  ;3251  Q_RC[06]  ; Get expected data
U 127D, 0001,203C,0180,09E0,0000,127E  ;3252  RC[0C]_Q  ; Save it for error logout
            ;3253  D_D.AND.RC[06],  ; Check for Bit 20 to be set
U 127E, 0811,0034,0180,0930,0010,127F  ;3254   CLK.UBCC ; Clock condition codes
U 127F, 0001,013C,0180,09D8,0000,10E0  ;3255  Z?,RC[0B]_D  ; ....
U 10E0, 0000,003C,0180,0800,0000,10E3  ;3256 =0 J/TST18.ERR.SUM  ; Error Summary bit ok
U 10E1, 0018,0038,0580,09C8,0000,10E2  ;3257  RC[9]_K[.1]  ; Set Flag for Fail Chain
```

```
            ;3258  .PAGE
            ;3259  ;
            ;3260  ;////////////////////////////////////////////////////////////////////////////////
            ;3261  ;
U 10E2. 0000.003D.D580.0800.0084.70A6  ;3262 =0 ERROR2[.6]  ; Error Summary Bit is clear should be set
            ;3263  ;
            ;3264  ;
            ;3265  ;////////////////////////////////////////////////////////////////////////////////
            ;3266  ;
            ;3267  ; ERROR LOGOUT:
            ;3268  ;
            ;3269  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3270  ;  CNFG REGISTER C/D ADDRESS
            ;3271  ;  EXPECTED MASKED REGISTER DATA
            ;3272  ;  RECIEVED MASKED REGISTER DATA
            ;3273  ;  ACTUAL RECIEVED REGISTER DATA
            ;3274  ;  NOT USED
            ;3275  ;
            ;3276  ;////////////////////////////////////////////////////////////////////////////////
            ;3277  ;
            ;3278  TST18.ERR.SUM:
U 10E3. 0010.0038.0180.0968.0200.1280  ;3279  VA_RC[0D]  ; Point to Cnfg Reg C/D
U 1280. 0818.0038.4180.0800.0000.1281  ;3280  D_k[.80]  ; Set up to clear the error bits
U 1281. 0000.003C.0180.A800.0000.1282  ;3281  CACHE.P_D[LONG]  ; Clear all the error bits
U 1282. 0003.003C.0180.09C8.0000.1283  ;3282  RC[9]_0  ; Delay, allow for the error bits
U 1283. 0000.003C.0180.0800.0000.1284  ;3283  nop  ; ... to decide whether to clear
U 1284. 0000.003C.0180.0800.0000.1285  ;3284  nop  ; ... or not
U 1285. 0000.003C.0180.A800.0000.1286  ;3285  CACHE.P_D[LONG]  ; Do a second write to the C/D register
            ;3286  ; ... to make sure they cleared
U 1286. 0000.003C.0180.C800.0000.1287  ;3287  D[LONG]_CACHE.P  ; Read back the Register C/D
U 1287. 0001.003C.0180.09D0.0000.1288  ;3288  RC[0A]_D  ; Get actual register data
U 1288. 0003.003C.0180.09E0.0000.1289  ;3289  RC[0C]_0  ; Expected Data
            ;3290  D_D.AND.RC[07],  ; Check to see if bits 31 and 7 are clear
U 1289. 0811.0034.0180.0938.0010.128A  ;3291  CLK.UBCC
U 128A. 0001.013C.0180.09D8.0000.10E4  ;3292  Z?,RC[0B]_D  ; Check Condition codes
```

```
            ;3293  .PAGE
            ;3294 ;
            ;3295 ;////////////////////////////////////////////////////////////////////////////////
            ;3296 ;
U 10E4, 0000,003D,D580,0800,0084,70A6  ;3297 =0 ERROR2[.6]  ; Error bits in Register C not clear
            ;3298 ;
            ;3299 ;
            ;3300 ;////////////////////////////////////////////////////////////////////////////////
            ;3301 ;
            ;3302 ; ERROR LOGOUT:
            ;3303 ;
            ;3304 ; DATA: I/O BASE ADDRESSS OF CURRENTLY TESTED MS780-E
            ;3305 ;  ADDRESS OF CNFG REGISTER C/D
            ;3306 ;  EXPECTED MASKED REGISTER DATA
            ;3307 ;  RECIEVED MASKED REGISTER DATA
            ;3308 ;  ACTUAL RECIEVED REGISTER DATA
            ;3309 ;  NOT USED
            ;3310 ;
            ;3311 ;////////////////////////////////////////////////////////////////////////////////
            ;3312 ;
U 10E5, 0010,0038,0180,0970,0200,128B  ;3313  VA_RC[0E]  ; Point to Cnfg Register A
            ;3314  D[LONG]_CACHE.P, ; Read it
U 128B, 0018,0038,0580,C9C8,0000,128C  ;3315   RC[9]_K[.1] ; Set flag for Fail Chain
U 128C, 0001,003C,0180,09D0,0000,128D  ;3316  RC[0A]_D  ; Get actual register data
U 128D, 0010,0038,01C0,0930,0000,128E  ;3317  Q_RC[06]          ; Get Mask
            ;3318  RC[0B]_D.AND.Q,  ; Mask bit 20
U 128E, 001D,0034,0180,09D8,0010,128F  ;3319   CLK.UBCC ; ....
U 128F, 0003,013C,0180,09E0,0000,10E6  ;3320  Z?,RC[0C]_0  ; Branch if error summary bit cleared
```

```
            ;3321  .PAGE
            ;3322 ;
            ;3323 ;////////////////////////////////////////////////////////////////////////////////
            ;3324 ;
U 10E6, 0000,003D,D580,0800,0084,70A6  ;3325 =0 ERROR2[.6]  ; Error summary bit did not clear
            ;3326 ;
            ;3327 ;
            ;3328 ;////////////////////////////////////////////////////////////////////////////////
            ;3329 ;
            ;3330 ; ERROR LOGOUT:
            ;3331 ;
            ;3332 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3333 ;  CNFG REGISTER C/D ADDRESS
            ;3334 ;  EXPECTED MASKED REGISTER DATA
            ;3335 ;  RECIEVED MASKED REGISTER DATA
            ;3336 ;  ACTUAL RECIEVED REGISTER DATA
            ;3337 ;  NOT USED
            ;3338 ;
            ;3339 ;////////////////////////////////////////////////////////////////////////////////
            ;3340 ;
U 10E7, 0000,003C,0180,0800,0000,1030  ;3341  J/RESTART.TEST.18 ; Go see if there are any more Ms780-E's
            ;3342 TEST.18.EXIT:   ; Exit test
U 1290, 0000,003C,0180,0800,0000,10E8  ;3343  NOP
            ;3344
```

```
;3345   .PAGE "TEST 1A6 D BUS PARITY CHECKING/GENERATION"
;3346  ;*************************************************************
;3347  ;+++
;3348  ;
;3349  ; Functional Description:
;3350  ; ----------------------
;3351  ;          This test verifies whether the controller can detect a parity error.
;3352  ;  This is done by initializing location 0 of the array  with some  test data.
;3353  ; When the test data is read back out, the Force D Bus Parity Error bit (CNFG
;3354  ; Reg B) is set thus forcing a Parity Error. Bits 20 and 19 (or 18 depending
;3355  ; on which controller is being addresed) in CNFG A register are checked to see
;3356  ; if they are set.
;3357  ;
;3358  ; SUBTEST 1 - PARITY GENERATION LOGIC
;3359  ;
;3360  ; Data patterns are written to the array and read back without forcing an
;3361  ; error.  If a parity error is reported then there is a fault in the parity
;3362  ; generation   logic or the error reporting logic.
;3363  ;
;3364  ; SUBTEST 2 - PARITY CHECKING LOGIC
;3365  ;
;3366  ; Data patterns are written to memory and is read out with the Force D Bus
;3367  ; Parity Error bit set.##
;3368  ; A check is made to see if bit 20 and 19 (or 18 as the case may be)
;3369  ; in CNFG A register operate correctly.
;3370  ;
;3371  ; The test patterns used in this test will be stored in cache and they are:
;3372  ;
;3373  ;
;3374  ;     00000000, BFBFBFBF, 76767676, ECECECEC, E5E5E5E5, D2D2D2D2, 99999999
;3375  ;
;3376  ;     8B8B8B8B, 10000000, 00000001, 00000100, 00010000, 11000000, 02313313
;3377  ;
;3378  ;     20131531, 44552922, 01010101, 01000101, 00010001, 00010100, 01010010
;3379  ;
;3380  ;     10101001, 11001011, 00111111, 00110100, 10100110, 01011101, 33224144
;3381  ;
;3382  ;     55445255, 89899489, 98988898
;3383  ;
;3384  ;
;3385  ;
;3386   ; CACHE DATA
;3387   ;%SO     ; Location 0 will get destroyed
;3388   ;$ 0000,0000,BFBF,BFBF,7676,7676,ECEC,ECEC
;3389   ;$ E5E5,E5E5,D2D2,D2D2,9999,9999,8B8B,8B8B
;3390   ;$ 0000,1000,0001,0000,0100,0000,0000,0001
;3391   ;$ 0000,1100,3313,0231,1531,2013,2922,4455
;3392   ;$ 0101,0101,0101,0100,0001,0001,0100,0001
;3393   ;$ 0010,0101,1001,1010,1011,1100,1111,0011
;3394   ;$ 0100,0011,0110,1010,1101,0101,4144,3322
;3395         ;$ 5255,5544,9489,8989,8898,9898
;3396
;3397  ;-------------------------------------------------------------
;3398  ;
;3399  ; Error Logout:
```

```
                ;3400 ; ------------
                ;3401 ;
                ;3402 ; See individual error reports.
                ;3403 ;
                ;3404 ;
                ;3405 ;=============================================================
                ;3406 ;
                ;3407 ; Register Map:
                ;3408 ; ------------
                ;3409 ;
                ;3410 ;     RA,RB     Description:
                ;3411 ;     -----     -----------
                ;3412 ;
                ;3413 ;       0       Pointer to Data in Cache
                ;3414 ;
                ;3415 ;
                ;3416 ;
                ;3417 ;      RC       Description:
                ;3418 ;      --       -----------
                ;3419 ;
                ;3420 ;       4       Error Summary Bit in CNFG Reg A (bit 20)
                ;3421 ;       5 Mask for bits <20:18> in CNFG Reg A
                ;3422 ;       6       Bit 11 in CNFG Reg B
                ;3423 ;       8 Recieved register contents
                ;3424 ;       9       Expected bit value upper/lower
                ;3425 ;       A See Indiviual Subtest
                ;3426 ;       B       See Indiviual Subtest
                ;3427 ;       C Bit pattern used by test
                ;3428 ;       D       Cntrl Register C/D Address
                ;3429 ;
                ;3430 ;
                ;3431 ;--
                ;3432 ;*************************************************************
                ;3433 =0
U 10E8, 0000,003D,0180,0800,0000,109B  ;3434 TST19: NEWTST
U 10E9, 0000,003C,0180,0800,0000,1034  ;3435  NOP
                ;3436 =00
U 1034, 0000,003D,0180,0800,0000,10BA  ;3437  CALL[FIND.MS780E]  ; Find all the MS780-E's
                ;3438       ; ... on the SBI
U 1035, 0000,003C,0180,0800,0000,12BE  ;3439  J/TEST.19.EXIT   ; No MS780-E's found
                ;3440 ;
                ;3441 ; Begin to set up the Test Constants
                ;3442 ;
U 1036, 0000,003C,2180,0800,0084,7037  ;3443  SC_K[.14]   ; Prepare to set up the Mask
U 1037, 0003,001C,0180,09A0,0000,1291  ;3444  RC[04]_NOT.MASK   ; Error summary bit in Cnfg
                ;3445       ; ...Reg A (bit 20) eqls 100000
U 1291, 0818,0038,4180,0800,0000,1292  ;3446  D_K[.80]   ; Mask for (bit 11) in
U 1292, 0000,003C,1180,0800,0084,7293  ;3447  SC_K[.4]
U 1293, 0D00,003C,0180,0800,0000,1294  ;3448  D_DAL.SC   ; ...CNFG Reg B
U 1294, 0001,003C,0180,09B0,0000,1295  ;3449  RC[06]_D   ; ...eqls 800
U 1295, 0000,003C,6580,0800,0084,7296  ;3450  SC_K[.10]   ; Mask for bits <20:18> in
U 1296, 0818,0038,8580,0800,0000,1297  ;3451  D_K[.C]    ; ...CNFG Reg A
U 1297, 0D00,003C,0180,0800,0000,1298  ;3452  D_DAL.SC   ; ...
U 1298, 0811,0030,0180,0920,0000,1299  ;3453  D_D.OR.RC[04]   ; ...eqls 1C0000
U 1299, 0001,003C,0180,09A8,0000,1038  ;3454  RC[05]_D   ; Save in RC[05]
```

```
            ;3455 =00
            ;3456 RESTART.TEST.19:    ; Entry point for second controller
U 1038, 0000.003D.0180.0800.0000.10AA  ;3457  CALL[START.TEST]  ; Configure the controllers
U 1039, 0000.003C.0180.0800.0000.12BE  ;3458  J/TEST.19.EXIT    ; No More controllers or MS780-E's
U 103A, 0000.003C.2180.0800.0084.703B  ;3459  SC_K[.14]    ; Set up bits for controllers UP/LO
U 103B, 0000.003C.0580.0800.0084.B29A  ;3460  SC_SC-1      ; ...
U 129A, 0018.0038.11C0.0800.0000.129B  ;3461  Q_K[.4]     ; Prepare to find out which controller
            ;3462        ; ... we are operating on
U 129B, 0000.003C.0180.0968.0000.129C  ;3463  LC_RC[0D]    ; Get address of CNFG A
            ;3464  Q_Q.AND.LC,    ; See if Bit 2 is set
U 129C, 0011.2034.01C0.0800.0010.129D  ;3465       CLK.UBCC    ; Check Condition Codes
U 129D, 0000.013C.0180.0800.0000.10EA  ;3466  Z?      ; ...
U 10EA, 0000.003C.0180.0800.0000.129E  ;3467 =0 J/TEST19.UPPER    ; If set we are operating on the UPPER
U 10EB, 0000.003C.0580.0800.0084.B29E  ;3468  SC_SC-1      ; Set up for lower controller
            ;3469 TEST19.UPPER:
U 129E, 0810.0038.0180.0920.0000.129F  ;3470  D_RC[04]    ; Set the status bits for CNFG A
U 129F, 0001.001C.0180.09C8.0000.10EC  ;3471  RC[09]_D.ORNOT.MASK  ; EQLS 180000 if UPPER Cntrl
            ;3472        ; ... or 140000 if LOWER Cntrl
            ;3473 ;
            ;3474 ;////////////////////////////////////////////////////////////////////////////////
            ;3475 ;
            ;3476 ;  SUBTEST 1
            ;3477 ;
U 10EC, 0000.003D.0180.0800.0000.11DB  ;3478 =0 SUBTEST     ; Start Subtest 1
U 10ED, 0018.0038.3580.0A80.0000.12A0  ;3479  R[0]_K[.50]    ; Set address of cache data
U 12A0, 0018.0038.8D80.09D8.0000.10EE  ;3480  RC[0B]_K[.1F]    ; Set the loop count
            ;3481 TST19.S1.LOOP:
U 10EE, 0000.003D.0180.0800.0000.11FE  ;3482 =0 CALL[ENABLE.CACHE]  ; Enable cache group 0
U 10EF, 0000.003C.0180.0A00.0200.12A1  ;3483  VA_R[0]     ; Get the Test Data
U 12A1, 0000.003C.0180.C800.0000.12A2  ;3484  D[LONG]_CACHE.P    ; Read the Pattern
U 12A2, 0001.003C.0180.09E0.0000.10F0  ;3485  RC[0C]_D    ; .... and save
U 10F0, 0000.003D.0180.0800.0000.11FA  ;3486 =0 CALL[DISABLE.CACHE]  ; Disable the cache
U 10F1, 0000.003C.0180.0800.0000.10F2  ;3487  nop     ; ...
U 10F2, 0000.003D.0180.0800.0000.11D3  ;3488 =0 ERLOOP     ; Set the Uaddress for loop on error
U 10F3, 0000.003C.0180.0800.0000.10F4  ;3489  nop     ; ...
U 10F4, 0000.003D.0180.0800.0000.10B4  ;3490 =0 CALL[SBI.INIT]    ; Initialize the SBI
U 10F5, 0000.003C.0180.0800.0000.10F6  ;3491  nop     ; ...
U 10F6, 0000.003D.0180.0800.0000.1259  ;3492 =0 CALL[MS780E.CLEANUP]  ; Clean up the Config Regs of MS780-E
U 10F7, 0003.003C.0180.0800.0200.12A3  ;3493  VA_ALU,ALU_0(A)    ; Set address of location Zero
U 12A3, 0810.0038.0180.0960.0000.12A4  ;3494  D_RC[0C]    ; Get Data  Pattern
U 12A4, 0000.003C.0180.A800.0000.12A5  ;3495  CACHE.P_D[LONG]    ; Write the Data pattern to the array
U 12A5, 0018.0038.1980.0800.0200.10F8  ;3496  VA_K[ZERO]    ; Point to location Zero
U 10F8, 0000.003D.8580.0800.0084.7216  ;3497 =0 SET.TRAP.MASK[.C]  ; Enable RDS Utraps
U 10F9, 0000.003C.0180.C800.0000.103C  ;3498  D[LONG]_CACHE.P    ; Read Back the Array
U 103C, 0000.003D.0180.0800.0000.1210  ;3499 =00 CHECK.UTRAP    ; Check for Utraps
U 103D, 0000.003C.0180.0800.0000.103F  ;3500  J/T19.S1.NOTRAP    ; Jump if no RDS trap
```

```
            ;3501   .PAGE
            ;3502 ;
            ;3503 ;////////////////////////////////////////////////////////////////////////////////
            ;3504 ;
U 103E, 0000.003D.0180.0800.0084.70A6  ;3505  ERROR2[.8]    ; RDS Utrap on normal read from array
            ;3506 ;
            ;3507 ;
            ;3508 ;////////////////////////////////////////////////////////////////////////////////
            ;3509 ;
            ;3510 ; ERROR LOGOUT:
            ;3511 ;
            ;3512 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3513 ;  NOT USED
            ;3514 ;  NOT USED
            ;3515 ;  NOT USED
            ;3516 ;  NOT USED
            ;3517 ;  NOT USED
            ;3518 ;  EXPECTED UTRAP FLAG
            ;3519 ;  RECIEVED UTRAP FLAG
            ;3520 ;
            ;3521 ;////////////////////////////////////////////////////////////////////////////////
            ;3522 ;
            ;3523 T19.S1.NOTRAP:
U 103F, 0010.0038.0180.0970.0200.12A6  ;3524  VA_RC[0E]    ; Point to CNFG Reg A
U 12A6, 0000.003C.0180.C800.0000.12A7  ;3525  D[LONG]_CACHE.P    ; Read the Register
U 12A7, 0001.003C.0180.09C0.0000.12A8  ;3526  RC[08]_D    ; Save it for error logout
            ;3527  D_D.AND.RC[05],    ; Check bits <20:18> for set
U 12A8, 0811.0034.0180.0928.0010.12A9  ;3528   clk.ubcc   ; Clock condition codes
U 12A9, 0003.013C.0180.09C8.0000.10FA  ;3529  Z?,RC[09]_0    ; Branch if bits are clear else..
```

```
            ;3530   .PAGE
            ;3531  ;
            ;3532  ;////////////////////////////////////////////////////////////////////////////////
            ;3533  ;
U 10FA, 0000.003D.5D80.0800.0084.70A6   ;3534 =0 ERROR2[.7]    ; None of bits <20:18> in Cnfg Reg A
            ;3535        ; ...should be set
            ;3536  ;
            ;3537  ;////////////////////////////////////////////////////////////////////////////////
            ;3538  ;
            ;3539  ; ERROR LOGOUT:
            ;3540  ;
            ;3541  ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3542  ;  NOT USED
            ;3543  ;  DATA WRITTEN TO ARRAY
            ;3544  ;  PATTERN COUNT
            ;3545  ;  NOT USED
            ;3546  ;  EXPECTED VALUE OF BITS <20:18> OF CNFG REG A
            ;3547  ;  RECIEVED REGISTER CONTENTS OF CNFG REG A
            ;3548  ;
            ;3549  ;////////////////////////////////////////////////////////////////////////////////
            ;3550  ;
U 10FB, 0019.0014.1180.0A80.0000.12AB   ;3551  R[0]_D+K[.4]    ; Select next test data
U 12AB, 0810.0038.0180.0958.0000.12AC   ;3552  D_RC[0B]    ; Get the loop count
            ;3553   RC[0B]_D-K[.1],    ; Done ?
U 12AC, 0019.0000.0580.09D8.0010.12AD   ;3554   CLK.UBCC
U 12AD, 0000.013C.0180.0800.0000.10FC   ;3555  Z?     ; Branch if yes
U 10FC, 0000.003C.0180.0800.0000.12A1   ;3556 =0 J/TST19.S1.LOOP          ; Try Next Pattern
U 10FD, 0000.003C.0180.0800.0000.10FE   ;3557  nop     ; ....
            ;3558  ;
            ;3559  ;////////////////////////////////////////////////////////////////////////////////
            ;3560  ;
            ;3561  ;  Subtest 2  Parity Checking logic
            ;3562  ;
            ;3563  ;////////////////////////////////////////////////////////////////////////////////
            ;3564  ;
U 10FE, 0000.003D.0180.0800.0000.11DB   ;3565 =0 SUBTEST     ; Increment the subtest number
U 10FF, 0018.0038.3580.0A80.0000.12AE   ;3566  R[0]_K[.50]    ; Point to data in cache
U 12AE, 0018.0038.8D80.09D8.0000.110A   ;3567  RC[0B]_K[.1F]    ; Set the loop count
            ;3568 TST19.S2.LOOP:
U 110A, 0000.003D.0180.0800.0000.11FE   ;3569 =0 CALL[ENABLE.CACHE]   ; Enable Cache Group 0
U 110B, 0000.003C.0180.0A00.0200.12AF   ;3570  VA_R[0]     ; Get the Test Data
U 12AF, 0000.003C.0180.C800.0000.12B0   ;3571  D[LONG]_CACHE.P    ; Read the Pattern
U 12B0, 0001.003C.0180.09E0.0000.1110   ;3572  RC[0C]_D    ; .... and save
U 1110, 0000.003D.0180.0800.0000.11FA   ;3573 =0 CALL[DISABLE.CACHE]   ; Disable the cache
U 1111, 0000.003C.0180.0800.0000.1112   ;3574  nop     ; ...
U 1112, 0000.003D.0180.0800.0000.11D3   ;3575 =0 ERLOOP     ; Set the Uaddress for loop on error
U 1113, 0000.003C.0180.0800.0000.1114   ;3576  nop     ; ...
U 1114, 0000.003D.0180.0800.0000.10B4   ;3577 =0 CALL[SBI.INIT]    ; Initialize the SBI
U 1115, 0000.003C.0180.0800.0000.1116   ;3578  nop     ; ...
U 1116, 0000.003D.0180.0800.0000.1259   ;3579 =0 CALL[MS780E.CLEANUP]   ; Clean up the Config Regs of MS780-E
U 1117, 0003.003C.0180.0800.0200.12B1   ;3580  VA_ALU,ALU_0(A)    ; Set address of location Zero
U 12B1, 0810.0038.0180.0960.0000.12B2   ;3581  D_RC[0C]    ; Get Data  Pattern
U 12B2, 0000.003C.0180.A800.0000.12B3   ;3582  CACHE.P_D[LONG]    ; Write the Data pattern to the array
U 12B3, 0010.0038.0180.0970.0200.12B4   ;3583  VA_RC[0E]    ; Set the Cnfg Address
U 12B4, 0810.0038.0180.0933.0000.1118   ;3584  VA_VA+4,D_RC[06]   ; Set the Force Parity Bit (bit 11)
```

```
          ;3585 =0 CACHE.P_D[LONG],  ; .... in Cnfg B
U 1118, 0000,003D,8580,A800,0084,7216  ;3586   SET.TRAP.MASK[.C] ; ... Enable RDS micro traps
U 1119, 0018,0038,1980,0800,0200,12B5  ;3587  VA_K[ZERO]   ; Point to location Zero
U 12B5, 0000,003C,0180,C800,0000,1040  ;3588  D[LONG]_CACHE.P   ; Read Back the Array
U 1040, 0000,003D,0180,0800,0000,1210  ;3589 =00 CHECK.UTRAP   ; Check for Utraps
U 1041, 0000,003C,0180,0800,0000,111A  ;3590  J/T.19.NTRP   ; Did not gget RDS Trap
U 1042, 0000,003C,0180,0800,0000,111B  ;3591  J/T.19.TRPOK   ; OK. Received RDS trap
          ;3592 =
```

```
            ;3593  .PAGE
            ;3594 ;
            ;3595 ;////////////////////////////////////////////////////////////////////////////////
            ;3596 ;
            ;3597 =0
            ;3598 T.19.NTRP:
U 111A, 0000,003D,0180,0800,0084,70A6  ;3599  ERROR2[.8]   ; Memory did not send RDS Tag
            ;3600 ;
            ;3601 ;////////////////////////////////////////////////////////////////////////////////
            ;3602 ;
            ;3603 ; ERROR LOGOUT:
            ;3604 ;
            ;3605 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3606 ;  NOT USED
            ;3607 ;  TEST PATTERN WRITTEN TO THE ARRAY
            ;3608 ;  LOOP COUNT
            ;3609 ;  NOT USED
            ;3610 ;  NOT USED
            ;3611 ;  EXPECTED UTRAP FLAG
            ;3612 ;  RECIEVED UTRAP FLAG
            ;3613 ;
            ;3614 ;////////////////////////////////////////////////////////////////////////////////
            ;3615 ;
            ;3616 T.19.TRPOK:
U 111B, 0010,0038,0180,0970,0200,12B6  ;3617  VA_RC[0E]   ; Point to Cnfg Reg A
            ;3618  D[LONG]_CACHE.P,  ; Read Register A
U 12B6, 0001,003C,0180,C9C0,0000,12B7  ;3619  RC[08]_D   ; ... and save it in RC[08]
U 12B7, 0811,0034,0180,0948,0000,12B8  ;3620  D_D.AND.RC[09]   ; Mask bits  <20:18>
            ;3621  ALU_D.XOR.RC[09],  ; Check the state of the error bits
U 12B8, 0011,0020,0180,0948,0010,12B9  ;3622   CLK.UBCC  ; Clock condition codes
U 12B9, 0001,013C,0180,09D0,0000,111C  ;3623  Z?,RC[0A]_D   ; Branch if bits OK
```

```
            ;3624  .PAGE
            ;3625 ;
            ;3626 ;////////////////////////////////////////////////////////////////////////////////
            ;3627 ;
U 111C, 0000,003D,5D80,0800,0084,70A6  ;3628 =0 ERROR2[.7]   ; Register A bits <20:18> incorrect
            ;3629 ;
            ;3630 ;
            ;3631 ;////////////////////////////////////////////////////////////////////////////////
            ;3632 ;
            ;3633 ; ERROR LOGOUT:
            ;3634 ;
            ;3635 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;3636 ;  NOT USED
            ;3637 ;  TEST PATTERN WRITTEN
            ;3638 ;  PATTERN COUNT
            ;3639 ;  RECIEVED MASKED REGISTER A CONTENTS
            ;3640 ;  EXPECTED MASKED REGISTER A CONTENTS
            ;3641 ;  RECIEVED ACTUAL REGISTER A CONTENTS
            ;3642 ;
            ;3643 ;////////////////////////////////////////////////////////////////////////////////
            ;3644 ;
U 111D, 0800,003C,0180,0A00,0000,12BA  ;3645  D_R[0]    ; Increment address of data in cache
U 12BA, 0019,0014,1180,0A80,0000,12BB  ;3646  R[0]_D+K[.4]   ; ....
U 12BB, 0810,0038,0180,0958,0000,12BC  ;3647  D_RC[0B]   ; Get the pattern count
            ;3648  RC[0B]_D-K[.1],
U 12BC, 0019,0000,0580,09D8,0010,12BD  ;3649   CLK.UBCC  ; Are we finished ?
U 12BD, 0000,013C,0180,0800,0000,111E  ;3650  Z?    ; Branch if done to exit
U 111E, 0000,003C,0180,0800,0000,12AF  ;3651 =0 J/TST19.S2.LOOP   ; Try Next Data Pattern
U 111F, 0000,003C,0180,0800,0000,1038  ;3652  J/RESTART.TEST.19  ; Go find next controller
            ;3653 TEST.19.EXIT:
U 12BE, 0000,003C,0180,0800,0000,1120  ;3654  NOP    ; FINISHED
            ;3655
```

```
          ;3656  .PAGE "TEST 1A7 ROM READ/WRITE ACCESS"
          ;3657 ;*************************************************************
          ;3658 ;***
          ;3659 ;
          ;3660 ; Functional Description:
          ;3661 ; ----------------------
          ;3662 ;
          ;3663 ; This test verifies that the ROM address space can be accessed for
          ;3664 ; READS, MAKSED WRITES, INTERLOCK WRITES and INTERLOCK READS.  The main
          ;3665 ; purpose of the WRITE tests is to see that no abnormal action takes
          ;3666 ; place from the SBI Interface (and/or Controller) that would hamper
          ;3667 ; normal CPU operation.
          ;3668 ;
          ;3669 ;
          ;3670 ;
          ;3671 ; Error Logout:
          ;3672 ; ------------
          ;3673 ;
          ;3674 ; See individual error reports.
          ;3675 ;
          ;3676 ;
          ;3677 ;
          ;3678 ;=============================================================
          ;3679 ;
          ;3680 ; Register Map:
          ;3681 ; ------------
          ;3682 ;
          ;3683 ;      RA,RB      Description:
          ;3684 ;      -----      -----------
          ;3685 ;
          ;3686 ; Not Used
          ;3687 ;
          ;3688 ;       RC        Description:
          ;3689 ;       --        -----------
          ;3690 ;
          ;3691 ; 4 Rom Found Flag
          ;3692 ;
          ;3693 ; B Loop Counter
          ;3694 ;
          ;3695 ; C Rom Address Pointer
          ;3696 ;
          ;3697 ; D I/O Base Address of Rom to be tested
          ;3698 ;
          ;3699 ; E I/O Base Address of MS780-E currently under test
          ;3700 ;
          ;3701 ; 8 Holds SBI.ERROR register when relevant
          ;3702 ;--
          ;3703 ;*************************************************************
          ;3704 =0
U 1120,  0000,003D,0180,0800,0000,109B  ;3705 T.20: NEWTST     ; Signify start of new test
U 1121,  0000,003C,0180,0800,0000,1044  ;3706  NOP
U 1044,  0000,003D,0180,0800,0000,10BA  ;3707 =00 CALL[FIND.MS780E]  ; Search out all MS780-E's on SBI
U 1045,  0000,003C,0180,0800,0000,12E8  ;3708  J/T.20.EXIT    ; No MS780-E's found - Exit
U 1046,  0003,003C,0180,09A0,0000,12BF  ;3709  RC[04]_0    ; Zero RC[04] which will be used to
          ;3710 =      ; signify Rom found(=1) or not(=0)
```

```
            ;3711 CHK.FOR.ROM:
            ;3712  ;
            ;3713  ; This is where we set up a value of 1000(hex) in the D reg and the Q
            ;3714  ; reg gets the base address of the MS780-E under test, then D and Q are
            ;3715  ; ORed together to form the base address for the ROM to be tested.
            ;3716  ;
U 12BF. 0000.003C.01F8.0800.0000.12C0  ;3717  Q_0     ; Load Q with zero's for shift
U 12C0. 0818.0038.0580.0800.0000.12C1  ;3718  D_K[.1]     ; Load D reg with a 1
U 12C1. 0000.003C.8580.0800.0084.72C2  ;3719  SC_K[.C]    ; Load SC for left shift of 12
U 12C2. 0D00.003C.0180.0800.0000.12C3  ;3720  D_DAL.SC    ; Shift D by SC so D = 1000(hex)
U 12C3. 0010.0038.01C0.0970.0000.12C4  ;3721  Q_RC[0E]    ; Load Q with MS780-E base address of
            ;3722       ; memory under test
U 12C4. 081D.0030.0180.0800.0000.12C5  ;3723  D_D.OR.Q    ; Form ROM Base Address
U 12C5. 0001.003C.0180.09E0.0000.12C6  ;3724  RC[0C]_D    ; Load RC[0C] with ROM base address
            ;3725  ;
            ;3726  ; Now we got the ROM base address, lets do a READ PHYSICAL on that
            ;3727  ; address to see if there is indeed a ROM out there.
            ;3728  ;
U 12C6. 0001.003C.0180.0800.0200.1122  ;3729  VA_D     ; Load VA with ROM base address
U 1122. 0000.003D.8980.0800.0084.7216  ;3730 =0 SET.TRAP.MASK[.D]  ; Enable Time-out traps
U 1123. 0000.003C.0180.C800.0000.12C7  ;3731  D[LONG]_CACHE.P    ; Attempt a read of the ROM
U 12C7. 0000.003C.0180.0800.0000.1048  ;3732  NOP
U 1048. 0000.003D.0180.0800.0000.1210  ;3733 =00 CHECK.UTRAP    ; Check for a micro trap
U 1049. 0000.003C.0180.0800.0000.12C8  ;3734  J/ROM.PRESENT    ; No, Time-out microtrap occured.
U 104A. 0000.003C.0180.0800.0000.1050  ;3735  NOP     ; Yes, Time-out microtrap occured.
            ;3736 =
            ;3737  ;
            ;3738  ; Fall through to here when a Time-out occured.  Check to see if there
            ;3739  ; are any more MS780-E's to check.  This routine will also be entered
            ;3740  ; from the routine that handles the case where there was NOT a micro
            ;3741  ; trap.
            ;3742 =00
            ;3743 ROM.NOT.PRESENT:
U 1050. 0000.003D.0180.0800.0000.124B  ;3744  CALL[ENABLE.NEXT.MS780E] ; Is there another MS780-E
U 1051. 0000.003C.0180.0800.0000.12BF  ;3745  J/CHK.FOR.ROM    ; Yes, go check for a rom
U 1052. 0000.003C.0180.0800.0000.12CF  ;3746  J/CHK.RC04    ; No, check to see if a ROM was found
            ;3747 =
            ;3748  ;
            ;3749  ; Come here when there was NO timeout to the ROM address.  If the ROM
            ;3750  ; found flag (RC[04]) is set then flag an error as there can be only
            ;3751  ; one rom, if it is clear then set it and go see if there are any more
            ;3752  ; MS780-E's to check.
            ;3753  ;
            ;3754 ROM.PRESENT:
U 12C8. 0010.0038.0180.0920.0010.12C9  ;3755  ALU_RC[04].CLK.UBCC  ; Move the Rom found flag to the ALU
U 12C9. 0000.013C.0180.0800.0000.1124  ;3756  Z?     ; Is the Rom found flag set?
U 1124. 0000.003C.0180.0800.0000.1126  ;3757 =0 J/TO.MANY.ROMS    ; Yes, go call an error
U 1125. 0000.003C.0180.0800.0000.12CA  ;3758  J/CONT.ROM.CHK    ; No, go set flag and continue
            ;3759 CONT.ROM.CHK:
U 12CA. 0018.0038.0580.09A0.0000.12CB  ;3760  RC[04]_K[.1]    ; Set Rom found flag
U 12CB. 0810.0038.0180.0970.0000.12CC  ;3761  D_RC[0E]    ; Save the I/O Base address of the
            ;3762       ; ...first memory with the ROMs enabled
U 12CC. 0001.003C.0180.09D8.0000.12CD  ;3763  RC[0B]_D    ; ...
U 12CD. 0810.0038.0180.0960.0000.12CE  ;3764  D_RC[0C]    ; Move Rom Base Address to the D reg
            ;3765  RC[0A]_D,    ; Save address of first rom found
```

```
U 12CE, 0001,003C,0180,09D0,0000,1050  ;3766   J/ROM.NOT.PRESENT
```

```
            ;3767  .PAGE
            ;3768 ;
            ;3769 ;////////////////////////////////////////////////////////////////////////////
            ;3770 ;
            ;3771 ; Report an error - more than one Rom was found (should only be one rom)
            ;3772 ;
            ;3773 =0
            ;3774 TO.MANY.ROMS:
U 1126, 0000,003D,8580,0800,0084,70A6  ;3775  ERROR2[.C]
            ;3776 ;
            ;3777 ;////////////////////////////////////////////////////////////////////////////
            ;3778 ;
            ;3779 ; ERROR LOGOUT:
            ;3780 ;
            ;3781 ; DATA: I/O Base Address of MS780-E currently under test
            ;3782 ;  I/O Base Address of First ROM Found
            ;3783 ;  I/O Base Address of MS780-E - ROM currently under test
            ;3784 ;  Unused
            ;3785 ;  Unused
            ;3786 ;  Unused
            ;3787 ;  Unused
            ;3788 ;  Unused
            ;3789 ;  Unused
            ;3790 ;  Unused
            ;3791 ;  Rom Found Flag  1 = ROM Found
            ;3792 ;     0 = No ROM Found
            ;3793 ;  Unused
            ;3794 ;
            ;3795 ;////////////////////////////////////////////////////////////////////////////
            ;3796 ;
U 1127, 0000,003C,0180,0800,0000,12CA  ;3797  J/CONT.ROM.CHK   ; On Continue, finish check for roms
            ;3798  ;
            ;3799  ; Come here after checking all MS780-E's that are configured in the
            ;3800  ; system for the presence of a ROM.  Check to see if a rom was found
            ;3801  ; (ie., RC[04] = 1 found rom, RC[04] = 0 did not find rom).
            ;3802  ;
            ;3803 CHK.RC04:
U 12CF, 0010,0038,0180,0920,0010,12D0  ;3804  ALU_RC[04],CLK.UBCC  ; Move Rom found flag to the ALU
U 12D0, 0000,013C,0180,0800,0000,1128  ;3805  Z?    ; Does flag = 1 or 0 ?
U 1128, 0000,003C,0180,0800,0000,112B  ;3806 =0 J/TEST.ROM   ; Flag is equal to 1, got test the ROM
U 1129, 0000,003C,0180,0800,0000,112A  ;3807  NOP    ; Flag is equal to 0, call an error
```

```
            ;3808  .PAGE
            ;3809 ;
            ;3810 ;////////////////////////////////////////////////////////////////////////////////
            ;3811 ;
            ;3812 ; No Rom present on system to test.  Call an error.
            ;3813 ;
U 112A, 0000,003D,8580,0800,0084,70A4  ;3814 =0 ERROR1[.C]
            ;3815 ;
            ;3816 ;////////////////////////////////////////////////////////////////////////////////
            ;3817 ;
            ;3818 ; ERROR LOGOUT:
            ;3819 ;
            ;3820 ; DATA: I/O Base Address of MS780-E currently under test
            ;3821 ;  CNFG C/D Address of MS780-E currently under test
            ;3822 ;  I/O Base Address of MS780-E - ROM currently under test
            ;3823 ;  Unused
            ;3824 ;  Unused
            ;3825 ;  Unused
            ;3826 ;  Unused
            ;3827 ;  Unused
            ;3828 ;  Unused
            ;3829 ;  Unused
            ;3830 ;  Rom Found Flag  1 = ROM Found
            ;3831 ;      0 = No ROM Found
            ;3832 ;  Unused
            ;3833 ;
            ;3834 ;////////////////////////////////////////////////////////////////////////////////
            ;3835 ;
            ;3836  ;
            ;3837  ; Come here when there is a ROM to test.  This is the main part of the
            ;3838  ; test.  Perform read/writes to rom and check for correct results.
            ;3839  ;
            ;3840 TEST.ROM:
U 112B, 0810,0038,0180,0950,0000,12D1  ;3841  D_RC[0A]    ; Get ROM Base Address for test
U 12D1, 0001,003C,0180,09E8,0000,12D2  ;3842  RC[0D]_D    ; Save in RC[0D]
U 12D2, 0001,003C,0180,09E0,0000,12D3  ;3843  RC[0C]_D    ; Save base address in RC[0C]
U 12D3, 0810,0038,0180,0958,0000,12D4  ;3844  D_RC[0B]    ; Restore I/O base address of MS780-E
            ;3845       ; ...under test
U 12D4, 0001,003C,0180,09F0,0000,12D5  ;3846  RC[0E]_D    ; ...
U 12D5, 0018,0038,8180,09D8,0000,12D6  ;3847  RC[0B]_K[.3FF]    ; RC[0B] gets loop count 3FF(hex)
U 12D6, 0010,0038,0180,0960,0200,112C  ;3848  VA_RC[0C]    ; Load VA with Rom base address
            ;3849  ;
            ;3850  ; This is the loop point that is used to start the Bus functions on the
            ;3851  ; next rom address.
            ;3852  ;
            ;3853 =0
            ;3854 TEST.LOOP:
U 112C, 0000,003D,0180,0800,0000,11D3  ;3855  ERLOOP     ; Loop on error from here
U 112D, 0000,003C,0180,0800,0000,112E  ;3856  NOP
U 112E, 0000,003D,0180,0800,0000,1084  ;3857 =0 CALL[SBI.INIT]    ; Initialize the SBI
U 112F, 0000,003C,0180,0800,0000,1130  ;3858  NOP
U 1130, 0000,003D,8980,0800,0084,7216  ;3859 =0 SET.TRAP.MASK[.D]  ; Enable Time-out traps
U 1131, 0000,003C,0180,0800,0000,1132  ;3860  NOP
U 1132, 0000,003D,8580,0800,0084,7216  ;3861 =0 SET.TRAP.MASK[.C]  ; Enable Read Data Substitute traps
            ;3862  ;
```

```
            ;3863   ; Perform a READ PHYSICAL
            ;3864   ;
U 1133. 0000.003C.0180.C800.0000.1054  ;3865  DT/LONG,MCT/READ.P  ; Perform a longword physical read
U 1054. 0000.003D.0180.0800.0000.1210  ;3866 =00 CHECK.UTRAP   ; Check to see if there was any traps
U 1055. 0000.003C.0180.0800.0000.1060  ;3867  J/CHK.READP.INT   ; No Time out, check for any interrupt
```

```
            ;3868  .PAGE
            ;3869 ;
            ;3870 ;////////////////////////////////////////////////////////////////////////////
            ;3871 ;
            ;3872 ; Got a Time Out on the Read Physical to the ROM, which should have happened.
            ;3873 ; Call an error.
            ;3874 ;
U 1056. 0000.003D.8580.0800.0084.70A6  ;3875  ERROR2[.C]
U 1057. 0000.003C.0180.0800.0000.1060  ;3876  NOP
            ;3877 ;
            ;3878 ;////////////////////////////////////////////////////////////////////////////
            ;3879 ;
            ;3880 ; ERROR LOGOUT:
            ;3881 ;
            ;3882 ; DATA: I/O Base Address of MS780-E currently under test
            ;3883 ;  I/O Base Address of First Rom Found
            ;3884 ;  Address of Rom location currently under test
            ;3885 ;  Loop Counter
            ;3886 ;  Unused
            ;3887 ;  Unused
            ;3888 ;  Expected Micro Trap flag
            ;3889 ;  Received Micro Trap flag
            ;3890 ;  Unused
            ;3891 ;  Unused
            ;3892 ;  Rom Found Flag  1 = ROM Found
            ;3893 ;      0 = No ROM Found
            ;3894 ;  Unused
            ;3895 ;
            ;3896 ;////////////////////////////////////////////////////////////////////////////
            ;3897 ;
            ;3898  ;
            ;3899  ; Check for any interrupts that came up during the Read Physical
            ;3900  ; Function.
            ;3901  ;
            ;3902 =00
            ;3903 CHK.READP.INT:
U 1060. 0000.003D.0180.0800.0000.1203  ;3904  CHECK.INTERRUPT   ; Any interrupts?
U 1061. 0000.003C.0180.0800.0000.1134  ;3905  J/DO.WRITEP   ; No, go perform the write
```

```
            ;3906  .PAGE
            ;3907 ;
            ;3908 ;////////////////////////////////////////////////////////////////////////////////
            ;3909 ;
            ;3910 ; Got and unexpected interrupt on the Read Physical function.  Call an error.
            ;3911 ;
U 1062, 0000.003D.8580.0800.0084.70A6  ;3912  ERROR2[.C]
U 1063, 0000.003C.0180.0800.0000.1134  ;3913  NOP
            ;3914 ;
            ;3915 ;////////////////////////////////////////////////////////////////////////////////
            ;3916 ;
            ;3917 ; ERROR LOGOUT:
            ;3918 ;
            ;3919 ; DATA: I/O Base Address of MS780-E currently under test
            ;3920 ;  I/O Base Address of First Rom Found
            ;3921 ;  Address of Rom location currently under test
            ;3922 ;  Loop Counter
            ;3923 ;  Unused
            ;3924 ;  Unused
            ;3925 ;  ID [vector] Register
            ;3926 ;  ID [sbi.err] Register
            ;3927 ;  ID [fault] Register
            ;3928 ;  Interrupt Flag  1 = Interrupt Occured
            ;3929 ;     0 = No Interrupt Occured
            ;3930 ;  Rom Found Flag  1 = ROM Found
            ;3931 ;     0 = No ROM Found
            ;3932 ;  Unused
            ;3933 ;
            ;3934 ;////////////////////////////////////////////////////////////////////////////////
            ;3935 ;
            ;3936  ;
            ;3937  ; Perform a WRITE PHYSICAL
            ;3938  ;
            ;3939 =0
            ;3940 DO.WRITEP:
U 1134, 0000.003D.0180.0800.0000.11D3  ;3941  ERLOOP    ; Loop on error from here
U 1135, 0000.003C.0180.0800.0000.1136  ;3942  NOP
U 1136, 0000.003D.0180.0800.0000.10B4  ;3943 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1137, 0000.003C.0180.0800.0000.1138  ;3944  NOP
U 1138, 0000.003D.8980.0800.0084.7216  ;3945 =0 SET.TRAP.MASK[.D]  ; Enable timeouts
U 1139, 0F00.003C.0180.0800.0000.12D7  ;3946  D_0     ; Zero the D register used for write
U 12D7, 0000.003C.0180.A800.0000.12D8  ;3947  DT/LONG,MCT/WRITE.P  ; Perform a Longword Physical write
U 12D8, 0000.003C.0180.0800.0084.72D9  ;3948  SC_K[.8]    ; Load SC with bit for mask operation
U 12D9, 0000.003C.65F0.2C00.0000.12DA  ;3949  Q_ID[SBI.ERR]    ; Put SBI Error reg in the Q reg
U 12DA, 0001.203C.0180.09C0.0000.12DB  ;3950  RC[08]_Q    ; Save for error log out
U 12DB, 0001.2024.0180.0800.0010.12DC  ;3951  ALU_Q.ANDNOT.MASK,CLK.UBCC ; Mask out bit 8
U 12DC, 0000.013C.0180.0800.0000.113A  ;3952  Z?     ; Is bit 8 set?
U 113A, 0000.003C.0180.0800.0000.113E  ;3953 =0 J/DO.INTREAD    ; yes, should be, continue with test
U 113B, 0000.003C.0180.0800.0000.113C  ;3954  NOP     ; No, should be, call an error
```

```
            ;3955  .PAGE
            ;3956 ;
            ;3957 ;////////////////////////////////////////////////////////////////////////////////
            ;3958 ;
            ;3959 ; Should have got a Error Confirmation (SBI.ERR bit <8>) from the Write
            ;3960 ; Physical but didn't.  Call an error.
            ;3961 ;
U 113C, 0000,003D,8580,0800,0084,70A6  ;3962 =0 ERROR2[.C]
U 113D, 0000,003C,0180,0800,0000,113E  ;3963  NOP
            ;3964 ;
            ;3965 ;////////////////////////////////////////////////////////////////////////////////
            ;3966 ;
            ;3967 ; ERROR LOGOUT:
            ;3968 ;
            ;3969 ; DATA: I/O Base Address of MS780-E currently under test
            ;3970 ;  I/O Base Address of First Rom Found
            ;3971 ;  Address of Rom location currently under test
            ;3972 ;  Loop Counter
            ;3973 ;  Unused
            ;3974 ;  Unused
            ;3975 ;  ID [sbi.err] Register
            ;3976 ;  Unused
            ;3977 ;  Unused
            ;3978 ;  Unused
            ;3979 ;  Rom Found Flag  1 = ROM Found
            ;3980 ;     0 = No ROM Found
            ;3981 ;  Unused
            ;3982 ;
            ;3983 ;////////////////////////////////////////////////////////////////////////////////
            ;3984 ;
            ;3985  ;
            ;3986  ; Perform a INTERLOCK READ
            ;3987  ;
            ;3988 =0
            ;3989 DO.INTREAD:
U 113E, 0000,003D,0180,0800,0000,11D3  ;3990  ERLOOP    ; Loop on error from here
U 113F, 0000,003C,0180,0800,0000,1140  ;3991  NOP
U 1140, 0000,003D,0180,0800,0000,10B4  ;3992 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1141, 0000,003C,0180,0800,0000,1142  ;3993  NOP
U 1142, 0000,003D,8980,0800,0084,7216  ;3994 =0 SET.TRAP.MASK[.D]  ; Enable Time-out traps
U 1143, 0000,003C,0180,0800,0000,1144  ;3995  NOP
U 1144, 0000,003D,8580,0800,0084,7216  ;3996 =0 SET.TRAP.MASK[.C]  ; Enable Read Data Substitute traps
U 1145, 0000,003C,0180,E800,0000,1064  ;3997  DT/LONG,MCT/LOCKREAD.P  ; Perform a Longword Interlock Read
U 1064, 0000,003D,0180,0800,0000,1210  ;3998 =00 CHECK.UTRAP   ; Any micro traps take place?
U 1065, 0000,003C,0180,0800,0000,1068  ;3999  J/CHK.INTREADP.INT  ; No, go check for any interrupts
```

```
            ;4000  .PAGE
            ;4001 ;
            ;4002 ;////////////////////////////////////////////////////////////////////////////////
            ;4003 ;
            ;4004 ; Got a Micro Trap (Time out) on the Longword Interlock Read Physical function.
            ;4005 ; Shouldn't have got this micro trap, call an error.
U 1066, 0000.003D.8580.0800.0084.70A6  ;4006  ERROR2[.C]
U 1067, 0000.003C.0180.0800.0000.1068  ;4007  NOP
            ;4008 ;
            ;4009 ;////////////////////////////////////////////////////////////////////////////////
            ;4010 ;
            ;4011 ; ERROR LOGOUT:
            ;4012 ;
            ;4013 ; DATA: I/O Base Address of MS780-E currently under test
            ;4014 ;  I/O Base Address of First Rom Found
            ;4015 ;  Address of Rom location currently under test
            ;4016 ;  Loop Counter
            ;4017 ;  Unused
            ;4018 ;  Unused
            ;4019 ;  Expected Micro Trap flag
            ;4020 ;  Received Micro Trap flag
            ;4021 ;  Unused
            ;4022 ;  Unused
            ;4023 ;  Rom Found Flag  1 = ROM Found
            ;4024 ;     0 = No ROM Found
            ;4025 ;  Unused
            ;4026 ;
            ;4027 ;////////////////////////////////////////////////////////////////////////////////
            ;4028 ;
            ;4029  ;
            ;4030  ; Check for any interrupts that came up during the Interlock Read
            ;4031  ; Physical Function.
            ;4032  ;
            ;4033 =00
            ;4034 CHK.INTREADP.INT:
U 1068, 0000.003D.0180.0800.0000.1203  ;4035  CHECK.INTERRUPT   ; Any interrupts?
U 1069, 0000.003C.0180.0800.0000.1146  ;4036  J/DO.INTWRITEP   ; No, go perform the interlock write
```

```
            ;4037  .PAGE
            ;4038 ;
            ;4039 ;////////////////////////////////////////////////////////////////////////////////
            ;4040 ;
            ;4041 ; Got and unexpected interrupt on the Interlock Read Physical function.  Call
            ;4042 ; an error.
            ;4043 ;
U 106A, 0000.003D.8580.0800.0084.70A6  ;4044  ERROR2[.C]
U 106B, 0000.003C.0180.0800.0000.1146  ;4045  NOP
            ;4046 ;
            ;4047 ;////////////////////////////////////////////////////////////////////////////////
            ;4048 ;
            ;4049 ; ERROR LOGOUT:
            ;4050 ;
            ;4051 ; DATA: I/O Base Address of MS780-E currently under test
            ;4052 ;  I/O Base Address of First Rom Found
            ;4053 ;  Address of Rom location currently under test
            ;4054 ;  Loop Counter
            ;4055 ;  Unused
            ;4056 ;  Unused
            ;4057 ;  ID [vector] Register
            ;4058 ;  ID [sbi.err] Register
            ;4059 ;  ID [fault] Register
            ;4060 ;  Interrupt Flag  1 = Interrupt Occured
            ;4061 ;     0 = No Interrupt Occured
            ;4062 ;  Rom Found Flag  1 = ROM Found
            ;4063 ;     0 = No ROM Found
            ;4064 ;  Unused
            ;4065 ;
            ;4066 ;////////////////////////////////////////////////////////////////////////////////
            ;4067 ;
            ;4068  ;
            ;4069  ; Perform INTERLOCK WRITE
            ;4070  ;
            ;4071 =0
            ;4072 DO.INTWRITEP:
U 1146, 0000.003D.0180.0800.0000.11D3  ;4073  ERLOOP    ; Loop on error from here
U 1147, 0000.003C.0180.0800.0000.1148  ;4074  NOP
U 1148, 0000.003D.0180.0800.0000.10B4  ;4075 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1149, 0000.003C.0180.0800.0000.114A  ;4076  NOP
U 114A, 0000.003D.8980.0800.0084.7216  ;4077 =0 SET.TRAP.MASK[.D]  ; Enable Timeouts
U 114B, 0F00.003C.0180.0800.0000.12DD  ;4078  D_0     ; Zero the D reg for the write
U 12DD, 0000.003C.0180.B800.0000.12DE  ;4079  DT/LONG,MCT/LOCKWRITE.P  ; Perfrom a Longword Interlock Write
U 12DE, 0000.003C.0180.0800.0084.72DF  ;4080  SC_K[.8]   ; Load SC with bit for mask operation
U 12DF, 0000.003C.65F0.2C00.0000.12E0  ;4081  Q_ID[SBI.ERR]   ; Put SBI Error reg in the Q reg
U 12E0, 0001.203C.0180.09C0.0000.12E1  ;4082  RC[08]_Q   ; Save for error logout
U 12E1, 0001.2024.0180.0800.0010.12E2  ;4083  ALU_Q.ANDNOT.MASK,CLK.UBCC ; Mask out bit 8
U 12E2, 0000.013C.0180.0800.0000.114C  ;4084  Z?    ; Is bit 8 set?
U 114C, 0000.003C.0180.0800.0000.114F  ;4085 =0 J/CHK.LOOP.CNT   ; yes, should be, continue with test
U 114D, 0000.003C.0180.0800.0000.114E  ;4086  NOP    ; No, should be, call an error
```

```
            ;4087   .PAGE
            ;4088  ;
            ;4089  ;////////////////////////////////////////////////////////////////////////////
            ;4090  ;
            ;4091  ; Should have got a Error Confirmation (SBI.ERR bit <8>) from the Interlock
            ;4092  ; Write Physical but didn't.  Call an error.
            ;4093  ;
U 114E, 0000,003D,8580,0800,0084,70A6   ;4094 =0 ERROR2[.C]
            ;4095  ;
            ;4096  ;////////////////////////////////////////////////////////////////////////////
            ;4097  ;
            ;4098  ; ERROR LOGOUT:
            ;4099  ;
            ;4100  ; DATA: I/O Base Address of MS780-E currently under test
            ;4101  ;  I/O Base Address of First Rom Found
            ;4102  ;  Address of Rom location currently under test
            ;4103  ;  Loop Counter
            ;4104  ;  Unused
            ;4105  ;  Unused
            ;4106  ;  ID [sbi.err] Register
            ;4107  ;  Unused
            ;4108  ;  Unused
            ;4109  ;  Unused
            ;4110  ;  Rom Found Flag  1 = ROM Found
            ;4111  ;     0 = No ROM Found
            ;4112  ;  Unused
            ;4113  ;
            ;4114  ;////////////////////////////////////////////////////////////////////////////
            ;4115  ;
            ;4116   ;
            ;4117   ; Decrement the loop count and check to see if we have made 1024(dec)
            ;4118   ; passes.
            ;4119   ;
            ;4120 CHK.LOOP.CNT:
U 114F, 0810,0038,0180,0958,0000,12E3   ;4121  D_RC[0B]    ; Load loop count in D reg
            ;4122  D_D-K[.1],    ; Subtract 1 from loop count
U 12E3, 0819,0000,0580,0800,0010,12E4   ;4123    CLK.UBCC   ; ...
U 12E4, 0001,013C,0180,09D8,0000,1150   ;4124  RC[0B]_D,Z?    ; Is loop count = 0 ?
U 1150, 0000,003C,0180,0800,0000,12E5   ;4125 =0 J/CONT.LOOP    ; No, continue with loop until 0
U 1151, 0000,003C,0180,0800,0000,12E8   ;4126  J/T.20.EXIT    ; yes, exit this test
            ;4127   ;
            ;4128   ; Come here when the loop count is Not Equal to 0.  Update RC[0C] and
            ;4129   ; VA with next ROM address, then continue with loop.
            ;4130   ;
            ;4131 CONT.LOOP:
U 12E5, 0810,0038,0180,0960,0000,12E6   ;4132  D_RC[0C]    ; Load Address in D for update
            ;4133  RC[0C]_D+K[.4],    ; Increment address by 4
U 12E6, 0019,0014,1180,09E0,0200,12E7   ;4134    VA_ALU    ; Store in VA for bus functions
U 12E7, 0000,003C,0180,0800,0000,112C   ;4135  J/TEST.LOOP    ; Go to begining of loop
            ;4136   ;
            ;4137   ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4138   ;
            ;4139 T.20.EXIT:
U 12E8, 0000,003C,0180,0800,0000,1152   ;4140  NOP     ; Exit the test
            ;4141
```

```
          ;4142
```

```
;4143  .PAGE "TEST 1A8 BYTE WRITE TO MEMORY TEST"
;4144 ;*************************************************************
;4145 ;+++
;4146 ;
;4147 ; Functional Description:
;4148 ; ----------------------
;4149 ;
;4150 ; This test writes a byte at a time to each of the four bytes within a
;4151 ; longword.  It starts out by writing zeroes to the memory, and consists
;4152 ; of rotating FF (hex) in a field of zeroes through each of the four
;4153 ; bytes.  Failure of this test will indicate malfunctions of the MASK
;4154 ; lines:  MASK Latch on the SBI Interface, MASK Latch and/or MASK Queue
;4155 ; on the Controller or the data paths between these.
;4156 ;
;4157 ;   [Subtest 1:]
;4158 ;
;4159 ; This subtest floats a byte of xFF or x00, as the case may be, across a
;4160 ; field of ones or zeroes.  The subtest initializes memory location zero
;4161 ; to some pattern and then writes a different pattern to the same
;4162 ; location.  These patters are as detailed below:
;4163 ;
;4164 ; Initial memory pattern  Byte Pattern for test
;4165 ; ----------------------  --------------------
;4166 ;
;4167 ;       xFFFFFF00         x000000FF
;4168 ;       xFFFF00FF         x0000FF00
;4169 ;       xFF00FFFF         x00FF0000
;4170 ;       x00FFFFFF         xFF000000
;4171 ;       x000000FF         xFFFFFF00
;4172 ;       x0000FF00         xFFFF00FF
;4173 ;       x00FF0000         xFF00FFFF
;4174 ;       xFF000000         x00FFFFFF
;4175 ;
;4176 ;   [Subtest 2:]
;4177 ;
;4178 ; This subtest initializes locations 0 and 4 to xFFFFFFFF and floats a
;4179 ; byte of zeroes across a quad-word field of ones.  It is expected that
;4180 ; only the addressed byte will be over-written.
;4181 ;
;4182 ;
;4183 ;
;4184 ; Error Logout:
;4185 ; ------------
;4186 ;
;4187 ; See individual error reports.
;4188 ;
;4189 ; Sync Point Description:
;4190 ; ----------------------
;4191 ;  N/A
;4192 ;
;4193 ;=============================================================
;4194 ;
;4195 ; Register Map:
;4196 ; ------------
;4197 ;
```

```
                ;4198 ;     RA,RB     Description:
                ;4199 ;     -----     -----------
                ;4200 ;
                ;4201 ;       0       Byte Pointer
                ;4202 ;
                ;4203 ;       1       Pass Counter
                ;4204 ;
                ;4205 ;       2 Memory location initialize pattern
                ;4206 ;
                ;4207 ;       3 Shift Data for the Q register
                ;4208 ;
                ;4209 ;      RC       Description:
                ;4210 ;      --       -----------
                ;4211 ; 3 Memory Test Pattern
                ;4212 ;
                ;4213 ; 4 Saved Passed counter
                ;4214 ;
                ;4215 ;       8 Address Pointer  Subtest 2
                ;4216 ;
                ;4217 ; 9 Address Pointer  Subtest 1
                ;4218 ;
                ;4219 ; A Memory Init pattern storage
                ;4220 ;
                ;4221 ; B Storage for memory data
                ;4222 ;
                ;4223 ; C Holds Expected Memory data
                ;4224 ;
                ;4225 ;--
                ;4226 ;*************************************************************
                ;4227 =0
U 1152, 0000,003D,0180,0800,0000,109B  ;4228 T.21: NEWTST    ; Signify start of new test
U 1153, 0000,003C,0180,0800,0000,106C  ;4229  NOP
U 106C, 0000,003D,0180,0800,0000,10BA  ;4230 =00 CALL[FIND.MS780E]  ; Search out all MS780-E's on the SBI
U 106D, 0000,003C,0180,0800,0000,134B  ;4231  J/T.21.EXIT   ; No MS780-E's to test, Exit this test
U 106E, 0000,003C,0180,0800,0000,1070  ;4232  NOP
                ;4233 =
                ;4234 =00
                ;4235 RESTART.TEST.21:
U 1070, 0000,003D,0180,0800,0000,10AA  ;4236  CALL[START.TEST]  ; Configure the MS780-E
U 1071, 0000,003C,0180,0800,0000,134B  ;4237  J/T.21.EXIT   ; Couldn't configure or No more MS780-E
                ;4238 ;
                ;4239 ;=============================================================================
                ;4240 ;
                ;4241 ; Subtest 1
                ;4242 ;
                ;4243 ;
                ;4244 T.21.SUB.1:
U 1072, 0000,003D,0180,0800,0000,11DB  ;4245  SUBTEST    ; Increment Subtest number
U 1073, 0003,003C,0180,09C8,0000,12E9  ;4246  RC[09]_0    ; Quad word location to be tested
U 12E9, 0018,0038,4980,09D0,0000,12EA  ;4247  RC[0A]_K[.FF]    ; Memory init pattern for pass 1
U 12EA, 0018,0038,4980,099B,0000,12EB  ;4248  RC[03]_K[.FF]    ; Memory test pattern for pass 1
U 12EB, 0000,003C,01F8,0800,0000,12EC  ;4249  Q_0     ; Zero the Q register
U 12EC, 0001,2028,0180,09E0,0000,12ED  ;4250  RC[0C]_NOT.Q   ; Move FFFFFFFF to RC[0C] - this is
                ;4251       ; the expected memory data for pass 1
U 12ED, 0001,2028,0180,0A98,0000,12EE  ;4252  R[3]_NOT.Q   ; Load R[3] with FFFFFFFF which will be
```

```
            ;4253       ; loaded in Q to shift one's into D
U 12EE, 0018,0038,0980,0A88,0000,12EF  ;4254  R[1]_K[.2]    ; Set pass counter for 2 passes
            ;4255   ;
            ;4256   ; LOOP.ONE will be re-entered after 4 byte writes of FF have been done
            ;4257   ; to the first and second longword.
            ;4258   ;
            ;4259 SUB1.LOOP.ONE:
U 12EF, 0003,003C,0180,0A80,0000,12F0  ;4260  R[0]_Q     ; Initialize the Byte pointer
U 12F0, 0000,003C,01C0,0A08,0000,12F1  ;4261  Q_R[1]     ; Move pass count to Q
U 12F1, 0001,203C,0180,09A0,0000,1154  ;4262  RC[04]_Q    ; Save pass count in RC[04] for error
            ;4263   ;
            ;4264   ; LOOP.TWO will be re-entered for each of the byte writes within a
            ;4265   ; longword.  After the four bytes within the longword have been tested
            ;4266   ; then the same is repeated for the second pass.
            ;4267   ;
            ;4268   ; NOTE:  The Init pattern for the memory test locations is just the
            ;4269   ;  one's complement of the test pattern.
            ;4270   ;
            ;4271 =0
            ;4272 SUB1.LOOP.TWO:
U 1154, 0000,003D,01B0,0800,0000,11D3  ;4273  ERLOOP     ; Loop on error from here
U 1155, 0000,003C,01B0,0800,0000,1156  ;4274  NOP
U 1156, 0000,003D,01B0,0800,0000,10B4  ;4275 =0 CALL[SBI.INIT]    ; Initialize the SBI
U 1157, 0000,003C,0180,0800,0000,1158  ;4276  NOP
U 1158, 0818,0039,0DB0,0800,0000,11EE  ;4277 =0 SET.DIAG.MODE[.3]  ; Set Diag Mode - ECC bypass
U 1159, 0003,003C,0180,0800,0200,12F2  ;4278  ALU_0(A),VA_ALU    ; Load VA with address 0
U 12F2, 0813,001C,0180,0950,0000,12F3  ;4279  D_NOT.RC[0A]    ; Invert the init pattern and move to D
U 12F3, 0000,003C,0180,A800,0000,12F4  ;4280  CACHE.P_D[LONG]    ; Write the first init pattern (loc. 0)
U 12F4, 0000,003C,0180,0803,0000,12F5  ;4281  VA_VA+4     ; Point VA to location 4
U 12F5, 0000,003C,01B0,A800,0000,12F6  ;4282  CACHE.P_D[LONG]    ; Write the second init pattern (loc.4)
U 12F6, 0000,003C,0180,0A00,0200,12F7  ;4283  VA_R[0]     ; Move test address (byte) to the VA
U 12F7, 0000,003C,01C0,0A00,0000,12F8  ;4284  Q_R[0]     ; Move test address to Q
U 12F8, 0810,0038,0180,0918,0000,12F9  ;4285  D_RC[03]    ; Move test pattern to the D register
            ;4286   ;
            ;4287   ; Do byte write physical to low longword.  The byte to be wrote is as
            ;4288   ; specified by the value of R[0] (ie., 0 = byte 0, 1 = byte, etc.).
            ;4289   ;
U 12F9, 0000,803C,01B0,A800,0000,1074  ;4290  CACHE.P_D[BYTE]    ; Write a byte to address by VA
U 1074, 0000,003D,01B0,0800,0000,1203  ;4291 =00 CHECK.INTERRUPT    ; Any Interrupts ?
U 1075, 0000,003C,01B0,0800,0000,115A  ;4292  J/DO.SEC.WRITE    ; No, do the second byte write
```

```
            ;4293  .PAGE
            ;4294 ;
            ;4295 ;////////////////////////////////////////////////////////////////////////////
            ;4296 ;
            ;4297 ; Got an unexpected interrupt on the byte write to the low longword.  Call an
            ;4298 ; error.
            ;4299 ;
U 1076, 0000.003D.8580.0800.0084.70A6  ;4300  ERROR2[.C]
U 1077, 0000.003C.0180.0800.0000.115A  ;4301  NOP
            ;4302 ;
            ;4303 ;////////////////////////////////////////////////////////////////////////////
            ;4304 ;
            ;4305 ; ERROR LOGOUT:
            ;4306 ;
            ;4307 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4308 ;  CNFG C/D Address of MS780-e Currently Under test
            ;4309 ;  NOT USED
            ;4310 ;  NOT USED
            ;4311 ;  NOT USED
            ;4312 ;  Quad Word Location Currently Under Test
            ;4313 ;  ID [vector] Register
            ;4314 ;  ID [sbi.err] Register
            ;4315 ;  ID [fault] Register
            ;4316 ;  Interrupt Occured Flag  1 = Interrupt Occured
            ;4317 ;       0 = No Interrupt Occured
            ;4318 ;  Pass Count
            ;4319 ;  NOT USED
            ;4320 ;
            ;4321 ;////////////////////////////////////////////////////////////////////////////
            ;4322 ;
            ;4323  ;
            ;4324  ; Perform a byte write physical to the second longword.
            ;4325  ;
            ;4326 =0
            ;4327 DO.SEC.WRITE:
U 115A, 0000.003D.0180.0800.0000.11D3  ;4328  ERLOOP    ; Loop on error from here
U 115B, 0000.003C.0180.0800.0000.115C  ;4329  NOP
U 115C, 0000.003D.0180.0800.0000.10B4  ;4330 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 115D, 0000.003C.0180.0800.0000.115E  ;4331  NOP
U 115E, 0B18.0039.0D80.0800.0000.11EE  ;4332 =0 SET.DIAG.MODE[.3]  ; Set diagnostic mode - ECC bypass
U 115F, 0B18.0030.1180.0A00.0000.12FA  ;4333  ALU_R[0].OR.K[.4],D_ALU  ; Set D to byte addr in 2nd longword
U 12FA, 0001.003C.0180.0800.0200.12FB  ;4334  VA_D    ; Load VA with address
U 12FB, 0B10.0038.0180.0918.0000.12FC  ;4335  D_RC[03]   ; Move test pattern to the D register
U 12FC, 0000.803C.0180.A800.0000.1078  ;4336  CACHE.P_D[BYTE]   ; Perfrom the byte write physical
U 1078, 0000.003D.0180.0800.0000.1203  ;4337 =00 CHECK.INTERRUPT   ; Any Interrupts ?
U 1079, 0000.003C.0180.0800.0000.1160  ;4338  J/DO.FIR.READ   ; No, do the read to check results
```

```
            ;4339  .PAGE
            ;4340 ;
            ;4341 ;////////////////////////////////////////////////////////////////////////////////
            ;4342 ;
            ;4343 ; Got an unexpected interrupt on the byte write to the second longword.  Call
            ;4344 ; an error.
            ;4345 ;
U 107A, 0000.003D.8580.0800.0084.70A6  ;4346  ERROR2[.C]
U 107B, 0000.003C.0180.0800.0000.1160  ;4347  NOP
            ;4348 ;
            ;4349 ;////////////////////////////////////////////////////////////////////////////////
            ;4350 ;
            ;4351 ; ERROR LOGOUT:
            ;4352 ;
            ;4353 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4354 ;  Unused
            ;4355 ;  NOT USED
            ;4356 ;  NOT USED
            ;4357 ;  NOT USED
            ;4358 ;  Quad Word Location Currently Under Test
            ;4359 ;  ID [vector] Register
            ;4360 ;  ID [sbi.err] Register
            ;4361 ;  ID [fault] Register
            ;4362 ;  Interrupt Occured Flag  1 = Interrupt Occured
            ;4363 ;       0 = No Interrupt Occured
            ;4364 ;  Pass Count
            ;4365 ;  NOT USED
            ;4366 ;
            ;4367 ;////////////////////////////////////////////////////////////////////////////////
            ;4368 ;
            ;4369  ;
            ;4370  ; This is where we start the read of the locations we just wrote ( 0
            ;4371  ; and 4) to check the write operation.  First, read the data at 0.
            ;4372  ;
            ;4373 =0
            ;4374 DO.FIR.READ:
U 1160, 0000.003D.0180.0800.0000.11D3  ;4375  ERLOOP    ; Loop on error from here
U 1161, 0000.003C.0180.0800.0000.1162  ;4376  NOP
U 1162, 0000.003D.0180.0800.0000.10B4  ;4377 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1163, 0000.003C.0180.0800.0000.1164  ;4378  NOP
U 1164, 0818.0039.0D80.0800.0000.11EE  ;4379 =0 SET.DIAG.MODE[.3]  ; Set diagnostic mode - ECC bypass
U 1165, 0003.003C.0180.0800.0200.12FD  ;4380  ALU_0(A),VA_ALU   ; Load VA with address 0
U 12FD, 0000.003C.0180.C800.0000.12FE  ;4381  D[LONG]_CACHE.P   ; Read longword location 0
U 12FE, 0001.003C.0180.09D8.0000.12FF  ;4382  RC[0B]_D   ; Save data in RC[0B]
U 12FF, 0011.0020.0180.0960.0010.1300  ;4383  ALU_D.XOR.RC[0C],CLK.UBCC ; Exclusive Or Expected & Received data
            ;4384  Z?,    ; Is the data read correct ?
U 1300, 0003.013C.0180.09A8.0000.1166  ;4385   RC[05]_0  ; Clear Interrupt Flag for Fail Chain
U 1166, 0000.003C.0180.0800.0000.1168  ;4386 =0 J/BAD.DATA.1   ; No, call an error
U 1167, 0000.003C.0180.0800.0000.1169  ;4387  J/DO.SEC.READ   ; Yes, go read second longword
```

```
            ;4388  .PAGE
            ;4389 ;
            ;4390 ;////////////////////////////////////////////////////////////////////////////////
            ;4391 ;
            ;4392 ; Come here if the data read from the first longword did not match the data
            ;4393 ; expected.  Call an error.
            ;4394 ;
            ;4395 =0
            ;4396 BAD.DATA.1:
U 1168, 0000,003D,8580,0800,0084,70A6  ;4397  ERROR2[.C]
            ;4398 ;
            ;4399 ;////////////////////////////////////////////////////////////////////////////////
            ;4400 ;
            ;4401 ; ERROR LOGOUT:
            ;4402 ;
            ;4403 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4404 ;  CNFG C/D ADDRESS OF MS780-E Currently Under Test
            ;4405 ;  Expected Memory Data
            ;4406 ;  Received Memory Data
            ;4407 ;  NOT USED
            ;4408 ;  Quad Word Location Currently Under Test
            ;4409 ;  NOT USED
            ;4410 ;  NOT USED
            ;4411 ;  NOT USED
            ;4412 ;  NOT USED
            ;4413 ;  NOT USED
            ;4414 ;  Pass Count
            ;4415 ;  NOT USED
            ;4416 ;
            ;4417 ;////////////////////////////////////////////////////////////////////////////////
            ;4418 ;
            ;4419  ;
            ;4420  ; Perform the Second read, data at location 4.
            ;4421  ;
            ;4422 DO.SEC.READ:
U 1169, 0000,003C,0180,0803,0000,1301  ;4423  VA_VA+4    ; Point VA to second longword
U 1301, 0000,003C,0180,C800,0000,1302  ;4424  D[LONG]_CACHE.P   ; Read the second longword
U 1302, 0001,003C,0180,09D8,0000,1303  ;4425  RC[0B]_D   ; Save data in RC[0B]
U 1303, 0011,0020,0180,0960,0010,1304  ;4426  ALU_D.XOR.RC[0C],CLK.UBCC ; Exclusive Or Expected & Received data
U 1304, 0000,013C,0180,0800,0000,116A  ;4427  Z?    ; Is the data read correct ?
U 116A, 0000,003C,0180,0800,0000,116C  ;4428 =0 J/BAD.DATA.2   ; No, call an error
U 116B, 0000,003C,0180,0800,0000,116D  ;4429  J/CHK.LOOP.CNT2   ; Yes, go check pass count
```

```
            ;4430  .PAGE
            ;4431 ;
            ;4432 ;////////////////////////////////////////////////////////////////////////////////
            ;4433 ;
            ;4434 ; Come here if the data read from the second longword did not match the data
            ;4435 ; expected.  Call an error.
            ;4436 ;
            ;4437 =0
            ;4438 BAD.DATA.2:
U 116C. 0000,003D,8580,0800,0084,70A6  ;4439  ERROR2[.C]
            ;4440 ;
            ;4441 ;////////////////////////////////////////////////////////////////////////////////
            ;4442 ;
            ;4443 ; ERROR LOGOUT:
            ;4444 ;
            ;4445 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4446 ;  Unused
            ;4447 ;  Expected Memory Data
            ;4448 ;  Received Memory Data
            ;4449 ;  NOT USED
            ;4450 ;  NOT USED
            ;4451 ;  NOT USED
            ;4452 ;  NOT USED
            ;4453 ;  NOT USED
            ;4454 ;  NOT USED
            ;4455 ;  NOT USED
            ;4456 ;  NOT USED
            ;4457 ;  BYTE WRITTEN
            ;4458 ;
            ;4459 ;////////////////////////////////////////////////////////////////////////////////
            ;4460 ;
            ;4461  ;
            ;4462  ; At this point we have done the two byte writes and reads.  Now update
            ;4463  ; the byte count (R[0]) and if = 4 then go check the pass count, else
            ;4464  ; shift the test pattern left by 8 with zero's if pass count = 2 OR
            ;4465  ; shift the test pattern left by 8 with one's if pass count = 1.
            ;4466  ;
            ;4467 CHK.LOOP.CNT2:
U 116D. 0800,003C,0180,0A00,0000,1305  ;4468  D_R[0]    ; Move byte count to the D register
U 1305. 0819,0014,0580,0800,0000,1306  ;4469  D_D+K[.1]   ; Increment the byte count
U 1306. 0001,003C,0180,0A80,0000,1307  ;4470  R[0]_D    ; Store byte count back in R[0]
U 1307. 0019,0020,1180,0800,0010,1308  ;4471  ALU_D.XOR.K[.4],CLK.UBCC ; Exclusive Or byte count with 4
U 1308. 0000,013C,0180,0800,0000,116E  ;4472  Z?     ; Is byte count = 4 ?
U 116E. 0000,003C,0180,0800,0000,1309  ;4473 =0 J/NOT.EQ.4   ; No, go shift and continue testing
U 116F. 0000,003C,0180,0800,0000,1314  ;4474  J/CHK.PASS.CNT.1  ; Yes, go check pass count
            ;4475  ;
            ;4476  ; Come here when byte count is NOT equal to 4.  Shift init pattern
            ;4477  ; left with the shift in value determined by the pass count.
            ;4478  ;
            ;4479 NOT.EQ.4:
U 1309. 0018,0034,0980,0A08,0010,130A  ;4480  ALU_R[1].AND.K[.2],CLK.UBCC ; AND pass count with 2
U 130A. 0000,013C,0180,0800,0000,1170  ;4481  Z?     ; Is pass count equal to 2 ?
U 1170. 0000,003C,0180,0800,0000,130F  ;4482 =0 J/SHIFT.ZEROS.IN  ; Yes, shift init pattern left w/zeros
U 1171. 0810,0038,0180,0950,0000,130B  ;4483  D_RC[0A]   ; No, shift init pattern left w/ones
U 130B. 0000,003C,01C0,0A18,0000,130C  ;4484  Q_R[3]    ; Load Q with FFFFFFFF for shift
```

```
U 130C, 0000,003C,0180,0800,0084,730D  ;4485  SC_K[.8]    ; Load SC with 8 for left shift
U 130D, 0D00,003C,0180,0800,0000,130E  ;4486  D_DAL.SC    ; Do shift
U 130E, 0001,003C,0180,09D0,0000,1154  ;4487  RC[0A]_D,J/SUB1.LOOP.TWO ; Re-load init pattern, then do
          ;4488       ; next loop
          ;4489 SHIFT.ZEROS.IN:
U 130F, 0000,003C,01F8,0800,0000,1310  ;4490  Q_0     ; Load Q with zero for shift
U 1310, 0810,0038,0180,0950,0000,1311  ;4491  D_RC[0A]    ; Load init pattern in the D register
U 1311, 0000,003C,0180,0800,0084,7312  ;4492  SC_K[.8]    ; load SC with +8 for left shift
U 1312, 0D00,003C,0180,0800,0000,1313  ;4493  D_DAL.SC    ; Do shift
U 1313, 0001,003C,0180,09D0,0000,1154  ;4494  RC[0A]_D,J/SUB1.LOOP.TWO ; Re-load init pattern, then do
          ;4495       ; next loop
          ;4496  ;
          ;4497  ; Come here to check the pass count.  Decrement the pass count and if
          ;4498  ; = 1 then do second pass, else if = 0 then go to next subtest.
          ;4499  ;
          ;4500 CHK.PASS.CNT.1:
U 1314, 0800,003C,0180,0A08,0000,1315  ;4501  D_R[1]     ; Move pass count to the D register
U 1315, 0019,0000,0580,0A88,0010,1316  ;4502  ALU_D-K[.1],R[1]_ALU,CLK.UBCC ; Decrement the pass count
U 1316, 0000,013C,0180,0800,0000,1172  ;4503  Z?      ; Is pass count = 0 ?
U 1172, 0000,003C,0180,0800,0000,1317  ;4504 =0 J/DO.SECOND    ; No, execute the second pass
U 1173, 0000,003C,0180,0800,0000,1174  ;4505  J/T.20.SUB.2    ; Yes, go to subtest 2
          ;4506  ;
          ;4507  ; Come here to set-up the Test pattern and Expected memory data for
          ;4508  ; the second pass.
          ;4509  ;
          ;4510 DO.SECOND:
U 1317, 0810,0038,0180,0960,0000,1318  ;4511  D_RC[0C]    ; Move FFFFFFFF to the D register
U 1318, 0001,003C,0180,0A98,0000,1319  ;4512  R[3]_D     ; Change R[3] to FFFFFFFF for shifting
          ;4513       ; on second pass
U 1319, 0019,0024,4980,09D0,0000,131A  ;4514  ALU_D.ANDNOT.K[.FF],RC[0A]_ALU ; Load FFFFFF00 into RC[0A], this is
          ;4515       ; the init pattern for pass two
U 131A, 0003,003C,0180,0998,0000,131B  ;4516  RC[03]_0    ; Memory test pattern for pass 2
U 131B, 0003,003C,0180,09E0,0000,12EF  ;4517  RC[0C]_0,J/SUB1.LOOP.ONE ; Load RC[0C] with Expected memory data
          ;4518       ; for pass two, then do second pass
          ;4519 ;
          ;4520 ;=============================================================================
          ;4521 ;
          ;4522 ; Subtest 2
          ;4523 ;
          ;4524 ;
          ;4525 =0
          ;4526 T.20.SUB.2:
U 1174, 0000,003D,0180,0800,0000,11DB  ;4527  SUBTEST     ; Increment Subtest number
U 1175, 0003,003C,0180,0A80,0000,131C  ;4528  R[0]_0      ; Init the pass count/byte count
U 131C, 0003,003C,0180,0A88,0000,131D  ;4529  R[1]_0      ; Longword pointer
U 131D, 0003,0028,0180,0A90,0000,131E  ;4530  R[2]_NOT.0    ; Load R[2] with init pattern: FFFFFFFF
U 131E, 0003,003C,0180,09C0,0000,131F  ;4531  RC[08]_0    ; Zero RC[08] for address pointer
U 131F, 0000,003C,01F8,0800,0000,1320  ;4532  Q_0     ; Zero the Q register
U 1320, 0001,2028,0180,09E0,0000,1321  ;4533  RC[0C]_NOT.Q    ; Load RC[0C] with FFFFFFFF, this is
          ;4534       ; the upper longword test pattern
U 1321, 0019,2024,4980,09D8,0000,1176  ;4535  ALU_Q.ANDNOT.K[.FF],RC[0B]_ALU ; Load FFFFFF00 into RC[0B], this is
          ;4536       ; the lower longword test pattern
          ;4537 =0
          ;4538 SUB2.LOOP.ONE:
U 1176, 0000,003D,0180,0800,0000,11D3  ;4539  ERLOOP     ; Loop on error from here
```

```
ZZ-ESKAR-V22  TEST 1A8 BYTE WRITE TO MEMORY TEST                                SEQ 0736
ESKAR45.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:23          Page  12

U 1177. 0000.003C.0180.0800.0000.1178  ;4540  NOP
U 1178. 0000.003D.0180.0800.0000.10B4  ;4541 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1179. 0000.003C.0180.0800.0000.117A  ;4542  NOP
U 117A. 0818.0039.0D80.0800.0000.11EE  ;4543 =0 SET.DIAG.MODE[.3]  ; Set diagnostic mode - ECC bypass
U 117B. 0003.003C.0180.0800.0200.1322  ;4544  ALU_0(A),VA_ALU   ; Load VA with address 0
U 1322. 0800.003C.0180.0A10.0000.1323  ;4545  D_R[2]    ; Move init pattern to the D register
U 1323. 0003.003C.0180.0800.0200.1324  ;4546  ALU_0(A),VA_ALU   ; Point VA to location 0
U 1324. 0000.003C.0180.A800.0000.1325  ;4547  CACHE.P_D[LONG]   ; Initialize location 0
U 1325. 0000.003C.0180.0803.0000.1326  ;4548  VA_VA+4    ; Update VA to point to location 4
U 1326. 0000.003C.0180.A800.0000.1327  ;4549  CACHE.P_D[LONG]   ; Initialize location 4
U 1327. 0003.003C.0180.0800.0200.1328  ;4550  ALU_0(A),VA_ALU   ; Point back to location 0
U 1328. 0810.0038.0180.0958.0000.1329  ;4551  D_RC[0B]   ; Load lower longword pattern in D
U 1329. 0000.003C.0180.A800.0000.132A  ;4552  CACHE.P_D[LONG]   ; Write the first longword
U 132A. 0000.003C.0180.0803.0000.132B  ;4553  VA_VA+4    ; Update VA to point to location 4
U 132B. 0810.0038.0180.0960.0000.132C  ;4554  D_RC[0C]   ; Load upper longword pattern in D
U 132C. 0000.003C.0180.A800.0000.132D  ;4555  CACHE.P_D[LONG]   ; Write the second longword
U 132D. 0000.003C.0180.C800.0000.132E  ;4556  D[LONG]_CACHE.P   ; Read second (upper) longword back
U 132E. 0001.003C.0180.09D0.0000.132F  ;4557  RC[0A]_D   ; Store in RC[0A]
U 132F. 0003.003C.0180.0800.0200.1330  ;4558  ALU_0(A),VA_ALU   ; Point to first longword
U 1330. 0000.003C.0180.C800.0000.1331  ;4559  D[LONG]_CACHE.P   ; Read first (lower) longword back
U 1331. 0001.003C.0180.09C8.0000.1332  ;4560  RC[09]_D   ; Store in RC[09]
U 1332. 0011.0020.0180.0958.0010.1333  ;4561  ALU_D.XOR.RC[0B],CLK.UBCC ; Exclusive Or expected & received data
U 1333. 0000.013C.0180.0800.0000.117C  ;4562  Z?    ; Is the data correct ?
U 117C. 0000.003C.0180.0800.0000.117E  ;4563 =0 J/BAD.DATA.3   ; No, call an error
U 117D. 0000.003C.0180.0800.0000.117F  ;4564  J/CHK.SECOND   ; Yes, go check the second longword
```

```
            ;4565  .PAGE
            ;4566 ;
            ;4567 ;////////////////////////////////////////////////////////////////////////////////
            ;4568 ;
            ;4569 ; Come here if the data read from the first longword did not match the data
            ;4570 ; expected.  Call an error.
            ;4571 ;
            ;4572 =0
            ;4573 BAD.DATA.3:
U 117E, 0000,003D,8580,0800,0084,70A6  ;4574  ERROR2[.C]
            ;4575 ;
            ;4576 ;////////////////////////////////////////////////////////////////////////////////
            ;4577 ;
            ;4578 ; ERROR LOGOUT:
            ;4579 ;
            ;4580 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4[illegible]   Unused
            ;45             Unused
            ;4583 ;  Expected Memory Data for Lower Longword
            ;4584 ;  Unused
            ;4585 ;  Received Memory Data for Lower Lonuword
            ;4586 ;  Address Pointer
            ;4587 ;  Unused
            ;4588 ;  Unused
            ;4589 ;  Unused
            ;4590 ;  Pass Count
            ;4591 ;  Unused
            ;4592 ;
            ;4593 ;////////////////////////////////////////////////////////////////////////////////
            ;4594 ;
            ;4595  ;
            ;4596  ; Now lets check the upper longword and see if it is correct.
            ;4597  ;
            ;4598 CHK.SECOND:
U 117F, 0810,0038,0180,0950,0000,1334  ;4599  D_RC[0A]   ; Load D with second longword read
U 1334, 0011,0020,0180,0960,0010,1335  ;4600  ALU_D.XOR.RC[0C],CLK.UBCC ; Exclusive Or expected & received data
U 1335, 0000,013C,0180,0800,0000,1180  ;4601  Z?    ; Is the data correct ?
U 1180, 0000,003C,0180,0800,0000,1182  ;4602 =0 J/BAD.DATA.4   ; No, call an error
U 1181, 0000,003C,0180,0800,0000,1183  ;4603  J/CHK.PASS.CNT.2  ; Yes, go check pass count
```

```
            ;4604  .PAGE
            ;4605 ;
            ;4606 ;////////////////////////////////////////////////////////////////////////////////
            ;4607 ;
            ;4608 ; Come here if the data read from the second longword did not match the data
            ;4609 ; expected.  Call an error.
            ;4610 ;
            ;4611 =0
            ;4612 BAD.DATA.4:
U 1182, 0000,003D,8580,0800,0084,70A6  ;4613  ERROR2[.C]
            ;4614 ;
            ;4615 ;////////////////////////////////////////////////////////////////////////////////
            ;4616 ;
            ;4617 ; ERROR LOGOUT:
            ;4618 ;
            ;4619 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4620 ;  Unused
            ;4621 ;  Expected Memory Data for Upper Longword
            ;4622 ;  Unused
            ;4623 ;  Received Memory Data for Upper Longword
            ;4624 ;  Unused
            ;4625 ;  Address Pointer
            ;4626 ;  Unused
            ;4627 ;  Unused
            ;4628 ;  Unused
            ;4629 ;  Pass Count
            ;4630 ;  Unused
            ;4631 ;
            ;4632 ;////////////////////////////////////////////////////////////////////////////////
            ;4633 ;
            ;4634  ;
            ;4635  ; Come here to check the pass count.  If pass count is less than 7,
            ;4636  ; then increment it and do next pass.  If pass count is equal to 7,
            ;4637  ; then restart this test to do the next controller.
            ;4638  ;
            ;4639 CHK.PASS.CNT.2:
U 1183, 0800,003C,0180,0A00,0000,1336  ;4640  D_R[0]    ; Load pass count in the D register
U 1336, 0019,0020,0D80,0800,0010,1337  ;4641  ALU_D.XOR.K[.3],CLK.UBCC ; Exclusive Or pass count with 3
U 1337, 0000,013C,0180,0800,0000,1184  ;4642  Z?    ; Is pass count equal to 3
U 1184, 0000,003C,0180,0800,0000,133C  ;4643 =0 J/CHK.PASS.CNT.3  ; No, finish with checking pass count
U 1185, 0018,0038,0580,0A88,0000,1338  ;4644  R[1]_K[.1]    ; Yes, point R[1] to second longword
U 1338, 0000,003C,01F8,0800,0000,1339  ;4645  Q_0    ; Zero the Q register
U 1339, 0001,2028,0180,09D8,0000,133A  ;4646  RC[0B]_NOT.Q    ; Change lower longword to FFFFFFFF
U 133A, 0019,2024,4980,09E0,0000,133B  ;4647  ALU_Q.ANDNOT.K[.FF],RC[0C]_ALU ; Change upper longword to FFFFFF00
U 133B, 0019,0014,0580,0A80,0000,1176  ;4648  R[0]_D+K[.1],J/SUB2.LOOP.ONE ; Increment the pass count and do
            ;4649       ; another pass.
            ;4650  ;
            ;4651  ; Come here when the pass count is not equal to three.  At the time
            ;4652  ; when the pass count is equal to three the 0 - byte will have to
            ;4653  ; be shifted up to the upper longword so when the count is 3 don't
            ;4654  ; take the normal path (this path) but execute the 5 previous micro
            ;4655  ; instructions which will swap the data as needed.
            ;4656  ;
            ;4657 CHK.PASS.CNT.3:
U 133C, 0800,003C,0180,0A00,0000,133D  ;4658  D_R[0]    ; Load pass count in the D register
```

```
U 133D, 0019,0020,5D80,0800,0010,133E  ;4659  ALU_D.XOR.K[.7],CLK.UBCC ; Exclusive Or pass count with 7
U 133E, 0000,013C,0180,0800,0000,1186  ;4660  Z?    ; Is pass count equal to 7
U 1186, 0000,003C,0180,0800,0000,133F  ;4661 =0 J/INC.PASS.CNT   ; No, go increment pass count and
          ;4662       ; continue with subtest 2
U 1187, 0000,003C,0180,0800,0000,1070  ;4663  J/RESTART.TEST.21  ; Yes, do next controller
          ;4664 INC.PASS.CNT:
U 133F, 0019,0014,0580,0A80,0000,1340  ;4665  R[0]_D+K[.1]   ; Increment the pass count and do
          ;4666       ; another pass.
U 1340, 0000,003C,0180,0A08,0010,1341  ;4667  ALU_R[1],CLK.UBCC  ; Move longword pointer to the ALU
U 1341, 0000,013C,0180,0800,0000,1188  ;4668  Z?    ; Is R[1] equal to 0 ?
U 1188, 0000,003C,0180,0800,0000,1346  ;4669 =0 J/INC.PASS.CNT.2  ; No, shift the upper longword
          ;4670  ;
          ;4671  ; This code will shift the lower longword
          ;4672  ;
U 1189, 0010,0038,01C0,0960,0000,1342  ;4673  Q_RC[0C]   ; Yes, shift the lower longword
U 1342, 0810,0038,0180,0958,0000,1343  ;4674  D_RC[0B]   ; Move lower longword to the D register
U 1343, 0000,003C,0180,0800,0084,7344  ;4675  SC_K[.8]   ; Load SC for left shift of 8
U 1344, 0D00,003C,0180,0800,0000,1345  ;4676  D_DAL.SC   ; Shift lower longword by 8
U 1345, 0001,003C,0180,09D8,0000,1176  ;4677  RC[0B]_D,J/SUB2.LOOP.ONE ; Restore in RC[0B] and continue test
          ;4678  ;
          ;4679  ; This code will shift the upper longword
          ;4680  ;
          ;4681 INC.PASS.CNT.2:
U 1346, 0010,0038,01C0,0958,0000,1347  ;4682  Q_RC[0B]   ; Move the lower longword to the Q reg
U 1347, 0810,0038,0180,0960,0000,1348  ;4683  D_RC[0C]   ; Move the upper longword to the D reg
U 1348, 0000,003C,0180,0800,0084,7349  ;4684  SC_K[.8]   ; Load SC for left shift of 8
U 1349, 0D00,003C,0180,0800,0000,134A  ;4685  D_DAL.SC   ; Shift upper longword by 8
U 134A, 0001,003C,0180,09E0,0000,1176  ;4686  RC[0C]_D,J/SUB2.LOOP.ONE ; Restore in RC[0C] and continue test
          ;4687  ;
          ;4688  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
          ;4689  ;
          ;4690 T.21.EXIT:
U 134B, 0000,003C,0180,0800,0000,118A  ;4691  NOP
          ;4692
          ;4693
```

```
;4694   .PAGE "TEST 1A9 UNALIGNED READ/WRITE FROM/TO  MEMORY"
;4695  ;*************************************************************
;4696  ;+++
;4697  ;
;4698  ; Functional Description:
;4699  ; ----------------------
;4700  ;
;4701  ; TEST 22 - UNALIGNED READS AND WRITES TO MEMORY -
;4702  ;
;4703  ;      This test checks the micro-trap resulting from an  unaligned
;4704  ; write  and  unaligned  read to/from memory.  The unaligned writes
;4705  ; are checked with longword reads on longword boundaries.  For  the
;4706  ; unaligned  read,  the test data is written using longword-aligned
;4707  ; writes.
;4708  ;
;4709  ;
;4710  ;
;4711  ;  SUBTEST 1 - UNALIGNED WRITE -
;4712  ;
;4713  ;      A test pattern will be stored in Cache.   This  pattern  is:
;4714  ; AAAAAAAA.
;4715  ; The test uses a byte counter to force the Unaligned  Data  Traps.
;4716  ; After  one memory reference the test pattern will be shifted left
;4717  ; 8 bits and the byte counter will be incremented  by  one.   Thus,
;4718  ; the data read back from memory (locations 0 and 4) without errors
;4719  ; should be:
;4720  ;
;4721  ;
;4722  ; Sequence Number of     Byte          Received Data from Location:
;4723  ; Unaligned Writes       Counter          0            4
;4724  ;
;4725  ;    First                 0          00000000     AAAAAAAA
;4726  ;
;4727  ;    Second                1          000000AA     AAAAAA00
;4728  ;
;4729  ;    Third                 2          0000AAAA     AAAA0000
;4730  ;
;4731  ;    Fourth                3          00AAAAAA     AA000000
;4732  ;
;4733  ;    Fifth                 4          AAAAAAAA     00000000
;4734  ;
;4735  ;
;4736  ;
;4737  ;  SUBTEST 2 - UNALIGNED READS -
;4738  ;
;4739  ;      Memory locations 0 and 4 will be initialize with a  specific
;4740  ; data pattern (same as in subtest 1).  A byte counter will be used
;4741  ; to perform unaligned reads  (READ.V.WCHK)  from  memory.   Checks
;4742  ; will be made for Unaligned uTraps and for data integrity.
;4743  ;
;4744  ;
;4745    ;
;4746    ; CACHE DATA
;4747    ;% 8
;4748    ;$ AAAA,AAAA
```

```
;4749  ;
;4750 ;
;4751 ; -----------------
;4752 ;
;4753 ; Error Logout:
;4754 ; ------------
;4755 ;
;4756 ; See individual error reports.
;4757 ;
;4758 ;
;4759 ;=============================================================
;4760 ;
;4761 ; Register Map:
;4762 ; ------------
;4763 ;
;4764 ;     RA,RB     Description:
;4765 ;     -----     -----------
;4766 ;
;4767 ;       0       ...
;4768 ;
;4769 ;       1       ...
;4770 ;
;4771 ;       :       :
;4772 ;       :       :
;4773 ;       :       :
;4774 ;
;4775 ;       E       ...
;4776 ;
;4777 ;       F       ...
;4778 ;
;4779 ;
;4780 ;      RC       Description:
;4781 ;      --       -----------
;4782 ;
;4783 ;       0       ...
;4784 ;
;4785 ;       1       ...
;4786 ;
;4787 ;       :       :
;4788 ;       :       :
;4789 ;       :       :
;4790 ;
;4791 ;       E       ...
;4792 ;
;4793 ;       F       ...
;4794 ;
;4795 ;--
;4796 ;*************************************************************
;4797 =0
U 118A, 0000,003D,0180,0800,0000,109B  ;4798 TST.22: NEWTST    ; Increment Test number
U 118B, 0000,003C,0180,0800,0000,107C  ;4799  NOP
U 107C, 0000,003D,0180,0800,0000,10BA  ;4800 =00 CALL[FIND.MS780E]  ; Look for MS780-E/Hs on the SBI
U 107D, 0000,003C,0180,0800,0000,1389  ;4801  J/TST.22.EXIT   ; Did not find any MS780-E/Hs
U 107E, 0000,003C,0180,0800,0000,118C  ;4802  NOP
;4803 =
```

```
U 118C, 0000,003D,0180,0800,0000,11FE  ;4804 =0 CALL[ENABLE.CACHE]  ; Enable cache to get the data
U 118D, 0018,0038,0180,0800,0200,134C  ;4805  VA_K[.8]   ; Point to the address of the data
        ;4806          ; ...in cache
U 134C, 0000,003C,0180,C800,0000,134D  ;4807  D[LONG]_CACHE.P   ; Read the cache
U 134D, 0001,003C,0180,0A80,0000,118E  ;4808  R[0]_D    ; Save in R[0]
U 118E, 0000,003D,0180,0800,0000,11FA  ;4809 =0 CALL[DISABLE.CACHE]  ; Disable the cache
U 118F, 0000,003C,0180,0800,0000,1080  ;4810  NOP
        ;4811 =00
        ;4812 RESTART.TST.22:    ; Entry point for next controller or
        ;4813          ; ...next MS780-E/H
U 1080, 0000,003D,0180,0800,0000,10AA  ;4814  CALL[START.TEST]  ; Configure the Controller
U 1081, 0000,003C,0180,0800,0000,1389  ;4815  J/TST.22.EXIT   ; Finished testing
U 1082, 0000,003C,0180,0800,0000,1190  ;4816  NOP
        ;4817 =
        ;4818 ;
        ;4819 ;////////////////////////////////////////////////////////////////////////////////
        ;4820 ;**
        ;4821 ;
        ;4822 ; SUBTEST 1
        ;4823 ;
        ;4824 ;--
        ;4825 ;////////////////////////////////////////////////////////////////////////////////
        ;4826 ;
U 1190, 0000,003D,0180,0800,0000,11DB  ;4827 =0 SUBTEST    ; Increment subtest number
U 1191, 0800,003C,0180,0A00,0000,134E  ;4828  D_R[0]    ; Get Test Pattern
U 134E, 0001,003C,0180,09D8,0000,134F  ;4829  RC[0B]_D   ; Save in RC[0B]
U 134F, 0003,003C,0180,09E0,0000,1350  ;4830  RC[0C]_0   ; Clear RC[0C]
U 1350, 0003,003C,0180,0A88,0000,1192  ;4831  R[1]_0    ; Set up a Byte Counter
        ;4832 =0
        ;4833 TST.22.LOOP1:
U 1192, 0000,003D,0180,0800,0000,11D3  ;4834  ERLOOP    ; Set up the Error looping mechanism
U 1193, 0000,003C,0180,0800,0000,1194  ;4835  NOP
U 1194, 0000,003D,0180,0800,0000,10B4  ;4836 =0 CALL[SBI.INIT]   ; Clear the SBI
U 1195, 0000,003C,0180,0800,0000,1196  ;4837  NOP
U 1196, 0818,0039,0D80,0800,0000,11EE  ;4838 =0 SET.DIAG.MODE[.3]  ; Select ECC Bypass Diagnostic Mode
        ;4839  VA_K[ZERO],    ; Point to location 0
U 1197, 0F18,0038,1980,0800,0200,1351  ;4840   D_0    ; Get data to initialize location 0
U 1351, 0000,003C,0180,A800,0000,1352  ;4841  CACHE.P_D[LONG]   ; Initailize location 0 to 0
U 1352, 0000,003C,0180,0803,0000,1353  ;4842  VA_VA+4    ; Point to location 4
U 1353, 0000,003C,0180,A800,0000,1354  ;4843  CACHE.P_D[LONG]   ; Initialize location 4 to 0
U 1354, 0000,003C,0180,0A08,0200,1198  ;4844  VA_R[1]    ; Get Byte Counter to VA (point to
        ;4845          ; ...the byte under test)
U 1198, 0000,003D,0580,0800,0084,7216  ;4846 =0 SET.TRAP.MASK[.1]  ; Enable Unaligned Data uTraps
U 1199, 0800,003C,0180,0A00,0000,1355  ;4847  D_R[0]    ; Get Data to be written to memory
        ;4848  MCT/WRITE.V.WCHK,DT/LONG, ; Force the Unaligned Write
U 1355, 0000,003C,0180,3000,0000,119A  ;4849   DK/NOP,VAK/NOP  ; ...
        ;4850 =0 D_K[.FF].LEFT,SI/MUL-,  ; Wait for the write to complete
U 119A, 0838,0039,4B80,0800,0000,1202  ;4851   CALL[WAIT.LOOP]  ; ...
U 119B, 0000,003C,0180,0800,0000,1084  ;4852  NOP
U 1084, 0000,003D,0180,0800,0000,1203  ;4853 =00 CHECK.INTERRUPT   ; Were there any unexpected Interrupts?
U 1085, 0000,003C,0180,0800,0000,1356  ;4854  J/TST.22.NOINTR1  ; OK. No interrupt occurred
```

```
            ;4855  .PAGE
            ;4856 ;
            ;4857 ;////////////////////////////////////////////////////////////////////////////////
            ;4858 ;
U 1086, 0000,003D,F580,0800,0084,70A6  ;4859  ERROR2[.A]    ; Unexpected Interrupt Occurred
            ;4860       ; ...on Unaligned Write
            ;4861 ;
            ;4862 ;////////////////////////////////////////////////////////////////////////////////
            ;4863 ;
            ;4864 ; ERROR LOGOUT:
            ;4865 ;
            ;4866 ; DATA: I/O BASE ADDRESS OF MS780-E CURRENTLY UNDER TEST
            ;4867 ;  ADDRESS OF CNFG REGISTER C/D OF CONTROLLER UNDER TEST
            ;4868 ;  NOT USED
            ;4869 ;  NOT USED
            ;4870 ;  NOT USED
            ;4871 ;  NOT USED
            ;4872 ;  ID VECTOR REGISTER
            ;4873 ;  ID SBI.ERR REGISTER
            ;4874 ;  ID FAULT REGISTER
            ;4875 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;4876 ;       0 = NO INTERRUPT OCCURRED )
            ;4877 ;
            ;4878 ;
            ;4879 ;////////////////////////////////////////////////////////////////////////////////
            ;4880 ;
U 1087, 0000,003C,0180,0800,0000,1356  ;4881  NOP
            ;4882 TST.22.NOINTR1:
U 1356, 0800,003C,0180,0A08,0010,1357  ;4883  D_R[1],CLK.UBCC    ; Get the Byte Counte and
            ;4884       ; ...see if it was 0
U 1357, 0000,013C,0180,0800,0000,119C  ;4885  Z?
            ;4886 =0 J/TST.22.NZ1,    ; Byte Counter is NOT 0
            ;4887   ALU_D.XOR.K[.4], ; See if the Byte Counter is 4
U 119C, 0019,0020,1180,0800,0010,1358  ;4888     CLK.UBCC ; ...
U 119D, 0000,003C,0180,0800,0000,1090  ;4889  J/TST.22.ZOR41    ; Byt Counter is 0
            ;4890 TST.22.NZ1:
U 1358, 0000,013C,0180,0800,0000,119E  ;4891  Z?
U 119E, 0000,003C,0180,0800,0000,1088  ;4892 =0 J/TST.22.NT41    ; Byte Counter is NOT 4
U 119F, 0000,003C,0180,0800,0000,1090  ;4893  J/TST.22.ZOR41    ; Byte Counter is 4
            ;4894 =00
            ;4895 TST.22.NT41:
U 1088, 0000,003D,0180,0800,0000,1210  ;4896  CHECK.UTRAP    ; See if Unaligned Trap Occurred
```

```
            ;4897  .PAGE
            ;4898 ;
            ;4899 ;////////////////////////////////////////////////////////////////////////////////
            ;4900 ;
            ;4901 TST.22.NOTRP1:
U 1089, 0000,003D,0180,0800,0084,70A4  ;4902  ERROR1[.B]   ; Unaligned Data Trap did NOT occur
            ;4903
            ;4904 ;
            ;4905 ;////////////////////////////////////////////////////////////////////////////////
            ;4906 ;
            ;4907 ; ERROR LOGOUT:
            ;4908 ;
            ;4909 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4910 ;  CNFG REG C/D ADDRESS OF CONTROLLER UNDER TEST
            ;4911 ;  NOT USED
            ;4912 ;  NOT USED
            ;4913 ;  NOT USED
            ;4914 ;  NOT USED
            ;4915 ;  EXPECTED TRAP FLAG
            ;4916 ;  RECEIVED TRAP FLAGS
            ;4917 ;
            ;4918 ;
            ;4919 ;////////////////////////////////////////////////////////////////////////////////
            ;4920 ;
            ;4921 TST.22.TRP1:
            ;4922  D_R[0],    ; Get data for 2nd reference
U 108A, 0800,003C,0180,0A03,0000,1359  ;4923   VA_VA+4   ; Bump up VA
            ;4924 =
            ;4925  MCT/WRITE.V.WCHK,MSC/SECOND.REF, ; Second referrence
U 1359, 0000,003C,0180,3000,3000,11A0  ;4926   DK/NOP,VAK/NOP
            ;4927 =0 D_K[.FF].LEFT,SI/MUL-,  ; Prepare to wait
U 11A0, 0838,0039,4B80,0800,0000,1202  ;4928   CALL[WAIT.LOOP]  ; ...
U 11A1, 0000,003C,0180,0800,0000,108C  ;4929  NOP
U 108C, 0000,003D,0180,0800,0000,1203  ;4930 =00 CHECK.INTERRUPT   ; Were there any unexpected Interrupts?
U 108D, 0000,003C,0180,0800,0000,1093  ;4931  J/TST.22.OK1   ; OK. No interrupt occurred
```

```
          ;4932  .PAGE
          ;4933 ;
          ;4934 ;////////////////////////////////////////////////////////////////////////////
          ;4935 ;
U 108E, 0000,003D,F580,0800,0084,70A6  ;4936  ERROR2[.A]   ; Unexpected Interrupt Occurred
          ;4937       ; ...on Unaligned Write
          ;4938 ;
          ;4939 ;////////////////////////////////////////////////////////////////////////////
          ;4940 ;
          ;4941 ; ERROR LOGOUT:
          ;4942 ;
          ;4943 ; DATA: I/O BASE ADDRESS OF MS780-E CURRENTLY UNDER TEST
          ;4944 ;  ADDRESS OF CNFG REGISTER C/D OF CONTROLLER UNDER TEST
          ;4945 ;  NOT USED
          ;4946 ;  NOT USED
          ;4947 ;  NOT USED
          ;4948 ;  NOT USED
          ;4949 ;  ID VECTOR REGISTER
          ;4950 ;  ID SBI.ERR REGISTER
          ;4951 ;  ID FAULT REGISTER
          ;4952 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
          ;4953 ;       0 = NO INTERRUPT OCCURRED )
          ;4954 ;
          ;4955 ;
          ;4956 ;////////////////////////////////////////////////////////////////////////////
          ;4957 ;
U 108F, 0000,003C,0180,0800,0000,1093  ;4958  J/TST.22.OK1   ; Skip the following code for
          ;4959       ; ..."CONTINUE" after ERROR call
          ;4960 =00
          ;4961 TST.22.ZOR41:
U 1090, 0000,003D,0180,0800,0000,1210  ;4962  CHECK.UTRAP   ; Location 0 or 4 was accessed.
          ;4963       ; ...No trap should be received.
U 1091, 0000,003C,0180,0800,0000,1093  ;4964  J/TST.22.OK1   ; OK. No uTrap was Received.
```

```
          ;4965  .PAGE
          ;4966 ;
          ;4967 ;////////////////////////////////////////////////////////////////////////////////
          ;4968 ;
U 1092, 0000,003D,0180,0800,0084,70A6  ;4969 =0 ERROR2[.8]   ; Unaligned Data Trap on referrence
          ;4970       ; ...to location 0 or 4
          ;4971 ;
          ;4972 ;////////////////////////////////////////////////////////////////////////////////
          ;4973 ;
          ;4974 ; ERROR LOGOUT:
          ;4975 ;
          ;4976 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;4977 ;  CNFG REG C/D ADDRESS OF CONTROLLER UNDER TEST
          ;4978 ;  NOT USED
          ;4979 ;  NOT USED
          ;4980 ;  NOT USED
          ;4981 ;  NOT USED
          ;4982 ;  EXPECTED TRAP FLAG
          ;4983 ;  RECEIVED TRAP FLAGS
          ;4984 ;
          ;4985 ;
          ;4986 ;////////////////////////////////////////////////////////////////////////////////
          ;4987 ;
          ;4988 TST.22.OK1:
U 1093, 0018,0038,1980,0800,0200,135A  ;4989  VA_K[ZERO]   ; Point to location 0
U 135A, 0000,003C,0180,C800,0000,135B  ;4990  D[LONG]_CACHE.P   ; Read location 0
          ;4991  RC[9]_D,   ; Save received data in RC[9]
U 135B, 0001,003C,0180,09CB,0000,135C  ;4992   VA_VA+4   ; Point to location 4
U 135C, 0000,003C,0180,C800,0000,135D  ;4993  D[LONG]_CACHE.P   ; Read location 4
U 135D, 0001,003C,0180,09D0,0000,135E  ;4994  RC[0A]_D   ; Save in RC[0A]
          ;4995  ALU_D.XOR.RC[0C],  ; Check if data received from the
U 135E, 0011,0020,0180,0960,0010,135F  ;4996   CLK.UBCC  ; ...upper location is equal to the
          ;4997       ; ...expected
U 135F, 0000,013C,0180,0800,0000,11A2  ;4998  Z?
```

```
          ;4999  .PAGE
          ;5000 ;
          ;5001 ;////////////////////////////////////////////////////////////////////////////////
          ;5002 ;
U 11A2, 0000,003D,D580,0800,0084,70A6  ;5003 =0 ERROR2[.6]   ; Upper Received data is NOT equal
          ;5004      ; ...to the Expected
          ;5005 ;
          ;5006 ;////////////////////////////////////////////////////////////////////////////////
          ;5007 ;
          ;5008 ; ERROR LOGOUT:
          ;5009 ;
          ;5010 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;5011 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
          ;5012 ;  EXPECTED DATA FROM UPPER LOCATION (LOCATION 4)
          ;5013 ;  EXPECTED DATA FROM LOWER LOCATION (LOCATION 0)
          ;5014 ;  RECEIVED DATA FROM UPPER LOCATION
          ;5015 ;  RECEIVED DATA FROM LOWER LOCATION
          ;5016 ;
          ;5017 ;
          ;5018 ;////////////////////////////////////////////////////////////////////////////////
          ;5019 ;
U 11A3, 0810,0038,0180,0948,0000,1360  ;5020  D_RC[9]    ; Get the lower received data
          ;5021  ALU_D.XOR.RC[0B],  ; Check to see if the Received data
U 1360, 0011,0020,0180,0958,0010,1361  ;5022   CLK.UBCC  ; ...from the lower location
U 1361, 0000,013C,0180,0800,0000,11A4  ;5023  Z?    ; ...is eqeul to the Expected
```

```
            ;5024  .PAGE
            ;5025 ;
            ;5026 ;////////////////////////////////////////////////////////////////////////////
            ;5027 ;
U 11A4, 0000,003D,D580,0800,0084,70A6  ;5028 =0 ERROR2[.6]   ; Upper Received data is NOT equal
            ;5029       ; ...to the Expected
            ;5030 ;
            ;5031 ;////////////////////////////////////////////////////////////////////////////
            ;5032 ;
            ;5033 ; ERROR LOGOUT:
            ;5034 ;
            ;5035 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5036 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
            ;5037 ;  EXPECTED DATA FROM UPPER LOCATION (LOCATION 4)
            ;5038 ;  EXPECTED DATA FROM LOWER LOCATION (LOCATION 0)
            ;5039 ;  RECEIVED DATA FROM UPPER LOCATION
            ;5040 ;  RECEIVED DATA FROM LOWER LOCATION
            ;5041 ;
            ;5042 ;
            ;5043 ;////////////////////////////////////////////////////////////////////////////
            ;5044 ;
U 11A5, 0800,003C,0180,0A08,0000,1362  ;5045  D_R[1]     ; Get the Byte Counter
            ;5046  ALU_D.XOR.K[.4],   ; Is the Byte Counter Equal to 4 yet?
U 1362, 0019,0020,1180,0800,0010,1363  ;5047   CLK.UBCC  ; ...
U 1363, 0000,013C,0180,0800,0000,11A6  ;5048  Z?
            ;5049 =0 J/TST.22.CONT1,   ; Not Yet. Continue with the test
U 11A6, 0019,0014,0580,0A88,0000,1364  ;5050   R[1]_D+K[.1]  ; Increment the Byte Counter
U 11A7, 0000,003C,0180,0800,0000,11A8  ;5051  J/TST.22.ENDS1   ; Byte Counter is 7. Finished Subtest 1
            ;5052 TST.22.CONT1:
            ;5053  D_RC[0B],   ; Prepare to shift the expected data
U 1364, 0810,0038,01F8,0958,0000,1365  ;5054   Q_0    ; ...8 bits to the left
U 1365, 0000,003C,0180,0800,0084,7366  ;5055  SC_K[.8]    ; ...
U 1366, 0D00,003C,0180,0800,0000,1367  ;5056  D_DAL.SC    ; Shift
U 1367, 0001,003C,0180,09D8,0000,1368  ;5057  RC[0B]_D    ; Save the shifted result
U 1368, 0810,0038,0180,0960,0000,1369  ;5058  D_RC[0C]    ; Get ready to shift now the upper LW
U 1369, 0000,003C,01C0,0A00,0000,136A  ;5059  Q_R[0]     ; ...
U 136A, 0000,003C,0180,0800,0084,736B  ;5060  SC_K[.8]    ; Must shift 8 bits over
U 136B, 0D00,003C,0180,0800,0000,136C  ;5061  D_DAL.SC    ; Shift
            ;5062  RC[0C]_D,   ; ...
U 136C, 0001,003C,0180,09E0,0000,1192  ;5063   J/TST.22.LOOP1  ; Continue with the test
            ;5064 ;
            ;5065 ;////////////////////////////////////////////////////////////////////////////
            ;5066 ;**
            ;5067 ;
            ;5068 ; SUBTEST 2
            ;5069 ;
            ;5070 ;--
            ;5071 ;////////////////////////////////////////////////////////////////////////////
            ;5072 ;
            ;5073 =0
            ;5074 TST.22.ENDS1:
U 11A8, 0000,003D,0180,0800,0000,11DB  ;5075  SUBTEST     ; Increment subtest number
U 11A9, 0800,003C,0180,0A00,0000,136D  ;5076  D_R[0]     ; Get Test data
U 136D, 0001,003C,0180,09D8,0000,136E  ;5077  RC[0B]_D    ; Save in RC[0B]
U 136E, 0003,003C,0180,09E0,0000,136F  ;5078  RC[0C]_0    ; Clear RC[0C]
```

```
U 136F. 0003.003C.0180.0A88.0000.11AA  ;5079  R[1]_0     ; Set up the Byte Counter
          ;5080 =0
          ;5081 TST.22.LOOP2:
U 11AA. 0000.003D.0180.0800.0000.11D3  ;5082  ERLOOP     ; Set up the Error looping mechanism
U 11AB. 0000.003C.0180.0800.0000.11AC  ;5083  NOP
U 11AC. 0000.003D.0180.0800.0000.10B4  ;5084 =0 CALL[SBI.INIT]   ; Clear the SBI
U 11AD. 0000.003C.0180.0800.0000.11AE  ;5085  NOP
U 11AE. 0818.0039.0D80.0800.0000.11EE  ;5086 =0 SET.DIAG.MODE[.3]  ; Set ECC Bypass Diagnostic Mode
U 11AF. 0018.0038.1980.0800.0200.1370  ;5087  VA_K[ZERO]    ; Point to location 1
U 1370. 0810.0038.0180.0958.0000.1371  ;5088  D_RC[0B]    ; Get data to initialize location 0
U 1371. 0000.003C.0180.A800.0000.1372  ;5089  CACHE.P_D[LONG]    ; Write data to location 0
          ;5090  D_RC[0C],    ; Get data to initialize location 4
U 1372. 0810.0038.0180.0963.0000.1373  ;5091    VA_VA+4    ; Point to location 4
U 1373. 0000.003C.0180.A800.0000.11B0  ;5092  CACHE.P_D[LONG]    ; Write the data to location 4
U 11B0. 0000.003D.0580.0800.0084.7216  ;5093 =0 SET.TRAP.MASK[.1]  ; Enable Unaligned Data Traps
U 11B1. 0000.003C.0180.0A08.0200.1374  ;5094  VA_R[1]     ; Point to the byte under test
          ;5095  MCT/READ.V.WCHK.DT/LONG, ; Force the unaligned trap
U 1374. 0000.003C.0180.5000.0000.1375  ;5096    DK/NOP.VAK/NOP  ; ...
U 1375. 0001.003C.0180.09C8.0000.1094  ;5097  RC[9]_D     ; Save the received data in RC[9]
U 1094. 0000.003D.0180.0800.0000.1203  ;5098 =00 CHECK.INTERRUPT    ; Were there any unexpected Interrupts?
U 1095. 0000.003C.0180.0800.0000.1376  ;5099  J/TST.22.NOINTR2  ; OK. No interrupt occurred
```

```
            ;5100  .PAGE
            ;5101 ;
            ;5102 ;////////////////////////////////////////////////////////////////////////////////
            ;5103 ;
U 1096. 0000,003D,F580,0800,0084,70A6  ;5104  ERROR2[.A]   ; Unexpected Interrupt Occurred
            ;5105      ; ...on Unaligned Read
            ;5106 ;
            ;5107 ;////////////////////////////////////////////////////////////////////////////////
            ;5108 ;
            ;5109 ; ERROR LOGOUT:
            ;5110 ;
            ;5111 ; DATA: I/O BASE ADDRESS OF MS780-E CURRENTLY UNDER TEST
            ;5112 ;  ADDRESS OF CNFG REGISTER C/D OF CONTROLLER UNDER TEST
            ;5113 ;  NOT USED
            ;5114 ;  NOT USED
            ;5115 ;  NOT USED
            ;5116 ;  NOT USED
            ;5117 ;  ID VECTOR REGISTER
            ;5118 ;  ID SBI.ERR REGISTER
            ;5119 ;  ID FAULT REGISTER
            ;5120 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;5121 ;       0 = NO INTERRUPT OCCURRED )
            ;5122 ;
            ;5123 ;
            ;5124 ;////////////////////////////////////////////////////////////////////////////////
            ;5125 ;
U 1097. 0000,003C,0180,0800,0000,1376  ;5126  NOP
            ;5127 TST.22.NOINTR2:
            ;5128  D_R[1],    ; Is the Byte Counter 0?
U 1376. 0800,003C,0180,0A08,0010,1377  ;5129   CLK.UBCC  ; ...
U 1377. 0000,013C,0180,0800,0000,11B2  ;5130  Z?
            ;5131 =0 J/TST.22.NZ2,   ; No. Byte Counter is NOT 0
            ;5132   ALU_D.XOR.K[.4], ; Is the Byte Counter 4?
U 11B2. 0019,0020,1180,0800,0010,1378  ;5133    CLK.UBCC ; ...
U 11B3. 0000,003C,0180,0800,0000,10A0  ;5134  J/TST.22.ZOR42   ; Byte Counter is 0.
            ;5135 TST.22.NZ2:
U 1378. 0000,013C,0180,0800,0000,11B4  ;5136  Z?
U 11B4. 0000,003C,0180,0800,0000,1098  ;5137 =0 J/TST.22.N42   ; Byte Counter is NOT 4
U 11B5. 0000,003C,0180,0800,0000,10A0  ;5138  J/TST.22.ZOR42   ; Byte Counter is 4
            ;5139 =00
            ;5140 TST.22.N42:
U 1098. 0000,003D,0180,0800,0000,1210  ;5141  CHECK.UTRAP   ; See if Unaligned Trap Occurred
```

```
            ;5142  .PAGE
            ;5143 ;
            ;5144 ;////////////////////////////////////////////////////////////////////////////
            ;5145 ;
            ;5146 TST.22.NOTRP2:
U 1099, 0000,003D,0180,0800,0084,70A4  ;5147  ERROR1[.8]   ; Unaligned Data Trap did NOT occur
            ;5148
            ;5149 ;
            ;5150 ;////////////////////////////////////////////////////////////////////////////
            ;5151 ;
            ;5152 ; ERROR LOGOUT:
            ;5153 ;
            ;5154 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;5155 ;  CNFG REG C/D ADDRESS OF CONTROLLER UNDER TEST
            ;5156 ;  NOT USED
            ;5157 ;  NOT USED
            ;5158 ;  NOT USED
            ;5159 ;  NOT USED
            ;5160 ;  EXPECTED TRAP FLAG
            ;5161 ;  RECEIVED TRAP FLAGS
            ;5162 ;
            ;5163 ;
            ;5164 ;////////////////////////////////////////////////////////////////////////////
            ;5165 ;
            ;5166 TST.22.TRP2:
            ;5167  VA_VA+4,   ; Bump up VA
U 109A, 0810,0038,0180,094B,0000,1379  ;5168   D_RC[9]   ; Get Data from first referrence
            ;5169 =
            ;5170
            ;5171  MCT/READ.V.WCHK,MSC/SECOND.REF, ; Second referrence
U 1379, 0000,003C,0180,5000,3000,137A  ;5172   DK/NOP,VAK/NOP  ; ...
U 137A, 0001,003C,0180,09C8,0000,109C  ;5173  RC[9]_D    ; Save the data returned
U 109C, 0000,003D,0180,0800,0000,1203  ;5174 =00 CHECK.INTERRUPT   ; Were there any unexpected Interrupts?
U 109D, 0000,003C,0180,0800,0000,10A3  ;5175  J/TST.22.OK2   ; OK. No interrupt occurred
```

```
            ;5176  .PAGE
            ;5177 ;
            ;5178 ;////////////////////////////////////////////////////////////////////////////
            ;5179 ;
U 109E, 0000,003D,F580,0800,0084,70A6  ;5180  ERROR2[.A]   ; Unexpected Interrupt Occurred
            ;5181       ; ...on Unaligned Read Second Ref.
            ;5182 ;
            ;5183 ;////////////////////////////////////////////////////////////////////////////
            ;5184 ;
            ;5185 ; ERROR LOGOUT:
            ;5186 ;
            ;5187 ; DATA: I/O BASE ADDRESS OF MS780-E CURRENTLY UNDER TEST
            ;5188 ;  ADDRESS OF CNFG REGISTER C/D OF CONTROLLER UNDER TEST
            ;5189 ;  NOT USED
            ;5190 ;  NOT USED
            ;5191 ;  NOT USED
            ;5192 ;  NOT USED
            ;5193 ;  ID VECTOR REGISTER
            ;5194 ;  ID SBI.ERR REGISTER
            ;5195 ;  ID FAULT REGISTER
            ;5196 ;  INTERRUPT OCCURRED FLAG ( 1 = INTERRUPT OCCURRED
            ;5197 ;       0 = NO INTERRUPT OCCURRED )
            ;5198 ;
            ;5199 ;
            ;5200 ;////////////////////////////////////////////////////////////////////////////
            ;5201 ;
U 109F, 0000,003C,0180,0800,0000,10A3  ;5202  J/TST.22.OK2   ; Skip the following code for
            ;5203       ; ..."CONTINUE" after ERROR call
            ;5204 =00
            ;5205 TST.22.ZOR42:
U 10A0, 0000,003D,0180,0800,0000,1210  ;5206  CHECK.UTRAP   ; Location 0 or 4 was accessed.
            ;5207       ; ...No trap should be received.
U 10A1, 0000,003C,0180,0800,0000,10A3  ;5208  J/TST.22.OK2   ; OK. No uTrap was Received.
```

```
          ;5209  .PAGE
          ;5210 ;
          ;5211 ;////////////////////////////////////////////////////////////////////////////////
          ;5212 ;
U 10A2, 0000,003D,0180,0800,0084,70A6  ;5213 =0 ERROR2[.8]   ; Unaligned Data Trap on referrence
          ;5214       ; ...to location 0 or 4
          ;5215 ;
          ;5216 ;////////////////////////////////////////////////////////////////////////////////
          ;5217 ;
          ;5218 ; ERROR LOGOUT:
          ;5219 ;
          ;5220 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;5221 ;  CNFG REG C/D ADDRESS OF CONTROLLER UNDER TEST
          ;5222 ;  NOT USED
          ;5223 ;  NOT USED
          ;5224 ;  NOT USED
          ;5225 ;  NOT USED
          ;5226 ;  EXPECTED TRAP FLAG
          ;5227 ;  RECEIVED TRAP FLAGS
          ;5228 ;
          ;5229 ;
          ;5230 ;////////////////////////////////////////////////////////////////////////////////
          ;5231 ;
          ;5232 TST.22.OK2:
U 10A3, 0800,003C,0180,0A00,0000,137B  ;5233  D_R[0]    ; Get expected data from Unaligned Read
          ;5234  ALU_D.XOR.RC[9],  ; Is it equal to the Received?
U 137B, 0011,0020,0180,0948,0010,137C  ;5235   CLK.UBCC  ; ...
          ;5236  Z?,
U 137C, 0001,013C,0180,09D0,0000,11B6  ;5237   RC[0A]_D  ; Save Expected Data for error logout
```

```
          ;5238  .PAGE
          ;5239 ;
          ;5240 ;////////////////////////////////////////////////////////////////////////////
          ;5241 ;
U 11B6, 0000,003D,D580,0800,0084,70A6  ;5242 =0 ERROR2[.6]   ; Data returned by Unaligned Read
          ;5243      ; ...is NOT equal to the expected
          ;5244 ;
          ;5245 ;////////////////////////////////////////////////////////////////////////////
          ;5246 ;
          ;5247 ; ERROR LOGOUT:
          ;5248 ;
          ;5249 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
          ;5250 ;  CNFG REGISTER C/D ADDRESS OF CONTROLLER UNDER TEST
          ;5251 ;  DATA WRITTEN TO LOCATION 4
          ;5252 ;  DATA WRITTEN TO LOCATION 0
          ;5253 ;  EXPECTED DATA FROM UNALIGNED READ
          ;5254 ;  RECEIVED DATA FROM UNALIGNED READ
          ;5255 ;
          ;5256 ;
          ;5257 ;////////////////////////////////////////////////////////////////////////////
          ;5258 ;
U 11B7, 0800,003C,0180,0A08,0000,137D  ;5259  D_R[1]    ; Get the Byte Counter
          ;5260  ALU_D.XOR.K[.4],  ; Is the Byte Counter Equal to 4 yet?
U 137D, 0019,0020,1180,0800,0010,137E  ;5261   CLK.UBCC  ; ...
U 137E, 0000,013C,0180,0800,0000,11B8  ;5262  Z?
          ;5263 =0 J/TST.22.CONT2,   ; Not Yet. Continue with the test
U 11B8, 0019,0014,0580,0A88,0000,137F  ;5264   R[1]_D+K[.1]  ; Increment the Byte Counter
U 11B9, 0000,003C,0180,0800,0000,1388  ;5265  J/TST.22.ENDS2   ; Byte Counter is 7. Finished Subtest 1
          ;5266 TST.22.CONT2:
          ;5267  D_RC[0B],   ; Prepare to shift the expected data
U 137F, 0810,0038,01F8,0958,0000,1380  ;5268   Q_0   ; ...8 bits to the left
U 1380, 0000,003C,0180,0800,0084,7381  ;5269  SC_K[.8]   ; ...
U 1381, 0D00,003C,0180,0800,0000,1382  ;5270  D_DAL.SC   ; Shift
U 1382, 0001,003C,0180,09D8,0000,1383  ;5271  RC[0B]_D   ; Save the shifted result
U 1383, 0810,0038,0180,0960,0000,1384  ;5272  D_RC[0C]   ; Prepare to shift upper LW
U 1384, 0000,003C,01C0,0A00,0000,1385  ;5273  Q_R[0]    ; ...
U 1385, 0000,003C,0180,0800,0084,7386  ;5274  SC_K[.8]   ; Must shift 8 bits
U 1386, 0D00,003C,0180,0800,0000,1387  ;5275  D_DAL.SC   ; Shift
          ;5276  RC[0C]_D,   ; Save shifted contents
U 1387, 0001,003C,0180,09E0,0000,11AA  ;5277   J/TST.22.LOOP2  ; Continue with the test
          ;5278 TST.22.ENDS2:
U 1388, 0000,003C,0180,0800,0000,1080  ;5279  J/RESTART.TST.22  ; Look for more controllers or
          ;5280      ; ...MS780-E/Hs
          ;5281 TST.22.EXIT:
U 1389, 0000,003C,0180,0800,0000,11BA  ;5282  NOP    ; Finished the test.
          ;5283
```

```
            ;5284  .PAGE "TEST 1AA RESERVED"
            ;5285 =0
U 11BA, 0000,003D,0180,0800,0000,109B  ;5286 T1AA: NEWTST
U 11BB, 0000,003C,0180,0800,0000,11BC  ;5287  NOP
            ;5288
```

```
            ;5289  .PAGE "TEST 1AB RESERVED"
            ;5290 =0
U 11BC. 0000,003D,0180,0800,0000,109B  ;5291 T1AB: NEWTST
U 11BD. 0000,003C,0180,0800,0000,11BE  ;5292  NOP
            ;5293
```

```
            ;5294  .PAGE "TEST 1AC RESERVED"
            ;5295 =0
U 11BE, 0000,003D,0180,0800,0000,109B  ;5296 T1AC: NEWTST
U 11BF, 0000,003C,0180,0800,0000,138A  ;5297  NOP
            ;5298
```

```
          ;5299  .PAGE "DUMMY NEWTST"
U 138A, 0000,003D,0180,0800,0000,109B  ;5300 DUM: NEWTST
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2755= 2756= 2757= 2760=
U 1008  2356= 2358= 2360= 2365= 2345= 2346= 2347= 2348=
U 1010  2379= 2381= 2382= 2386= 2402= 2403= 2404= 2412=
U 1018  2882= 2883= 2884= 2885= 2886= 2887= 2888= 2889=
U 1020  2414= 2416= 2418= 2419= 2462= 2463= 2464= 2466=
U 1028  2510= 2511= 2512= 2514= 3195= 3197= 3198= 3199=
U 1030  3207= 3208= 3209= 1902 3437= 3439= 3443= 3444=
U 1038  3457= 3458= 3459= 3460= 3499= 3500= 3505= 3524=
U 1040  3589= 3590= 3591= 1903 3707= 3708= 3709= 1904
U 1048  3733= 3734= 3735= 1906 1912= 1913= 2021: 1907
U 1050  3744= 3745= 3746= 1914 3866= 3867= 3875= 3876=
U 1058  1965= 1967= 2067: 1915 1978= 1979= 2101: 1916
U 1060  3904= 3905= 3912= 3913= 3998= 3999= 4006= 4007=
U 1068  4035= 4036= 4044= 4045= 4230= 4231= 4232= 1917
U 1070  4236= 4237= 4245= 4246= 4291= 4292= 4300= 4301=
U 1078  4337= 4338= 4346= 4347= 4800= 4801= 4802= 1918
U 1080  4814= 4815= 4816= 1919 4853= 4854= 4859= 4881=
U 1088  4896= 4902= 4923= 1920 4930= 4931= 4936= 4958=
U 1090  4962= 4964= 4969= 4989= 5098= 5099= 5104= 5126=
U 1098  5141= 5147= 5168= 1954 5174= 5175= 5180= 5202=
U 10A0  5206= 5208= 5213= 5233= 2039= 2040= 2085= 2086=
U 10A8  2208= 2209= 2334= 2335= 2337= 2342= 2572= 2577=
U 10B0  2582= 2587= 2592= 2593= 2637= 2638= 2715= 2716=
U 10B8  2794= 2795= 2861= 2862= 2870= 2871= 2872= 2874=
U 10C0  2877= 2878= 2919= 2920= 2921= 2922= 2929= 2930=
U 10C8  2936= 2938= 2941= 2942= 2961= 2962= 2964= 2965=
U 10D0  3054= 3055= 3058= 3059= 3070= 3071= 3074= 3075=
U 10D8  3192= 3193= 3211= 3212= 3213= 3214= 3232= 3248=
U 10E0  3256= 3257= 3262= 3279= 3297= 3313= 3325= 3341=
U 10E8  3434= 3435= 3467= 3468= 3478= 3479= 3482= 3483=
U 10F0  3486= 3487= 3488= 3489= 3490= 3491= 3492= 3493=
U 10F8  3497= 3498= 3534= 3551= 3556= 3557= 3565= 3566=
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895: 1955 3569= 3570= 1896: 1897: 1898: 1899:
U 1110  3573= 3574= 3575= 3576= 3577= 3578= 3579= 3580=
U 1118  3586= 3587= 3599= 3617= 3628= 3645= 3651= 3652=
U 1120  3705= 3706= 3730= 3731= 3757= 3758= 3775= 3797=
U 1128  3806= 3807= 3814= 3841= 3855= 3856= 3857= 3858=
U 1130  3859= 3860= 3861= 3865= 3941= 3942= 3943= 3944=
U 1138  3945= 3946= 3953= 3954= 3962= 3963= 3990= 3991=
U 1140  3992= 3993= 3994= 3995= 3996= 3997= 4073= 4074=
U 1148  4075= 4076= 4077= 4078= 4085= 4086= 4094= 4121=
U 1150  4125= 4126= 4228= 4229= 4273= 4274= 4275= 4276=
U 1158  4277= 4278= 4328= 4329= 4330= 4331= 4332= 4333=
U 1160  4375= 4376= 4377= 4378= 4379= 4380= 4386= 4387=
U 1168  4397= 4423= 4428= 4429= 4439= 4468= 4473= 4474=
U 1170  4482= 4483= 4504= 4505= 4527= 4528= 4539= 4540=
U 1178  4541= 4542= 4543= 4544= 4563= 4564= 4574= 4599=
U 1180  4602= 4603= 4613= 4640= 4643= 4644= 4661= 4663=
U 1188  4669= 4673= 4798= 4799= 4804= 4805= 4809= 4810=
U 1190  4827= 4828= 4834= 4835= 4836= 4837= 4838= 4840=
U 1198  4846= 4847= 4851= 4852= 4888= 4889= 4892= 4893=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   4928= 4929= 5003= 5020= 5028= 5045= 5050= 5051=
U 11A8   5075= 5076= 5082= 5083= 5084= 5085= 5086= 5087=
U 11B0   5093= 5094= 5133= 5134= 5137= 5138= 5242= 5259=
U 11B8   5264= 5265= 5286= 5287= 5291= 5292= 5296= 5297=
U 11C0   1956 1957 1958 1959 1960 1961 1962 1963
U 11C8   1964 1968 1969 1970 1971 1972 1973 1974
U 11D0   1975 1976 1977 1996 2019 2020 2065 2066
U 11D8   2149 2150 2151 2169 2171 2197 2198 2199
U 11E0   2200 2201 2202 2203 2204 2206 2207 2343
U 11E8   2344 2369 2389 2390 2391 2398 2538 2539
U 11F0   2540 2541 2542 2543 2567 2615 2639 2640
U 11F8   2641 2642 2660 2661 2662 2663 2681 2682
U 1200   2683 2684 2710 2744 2746 2747 2748 2750
U 1208   2751 2752 2753 2754 2761 2762 2763 2764
U 1210   2787 2788 2789 2790 2792 2793 2822 2823
U 1218   2824 2863 2864 2865 2866 2875 2876 2880
U 1220   2881 2894 2895 2897 2898 2899 2901 2906
U 1228   2907 2908 2910 2915 2917 2918 2926 2927
U 1230   2931 2932 2933 2934 2935 2943 2944 2945
U 1238   2947 2948 2949 2950 2951 2952 2957 2959
U 1240   2960 2963 2988 2989 2990 2991 3019 3020
U 1248   3022 3023 3024 3050 3052 3053 3057 3061
U 1250   3062 3063 3064 3066 3067 3068 3069 3072
U 1258   3073 3121 3123 3124 3125 3127 3129 3130
U 1260   3132 3134 3135 3136 3137 3140 3141 3142
U 1268   3143 3145 3200 3201 3202 3203 3215 3216
U 1270   3217 3218 3219 3220 3221 3222 3223 3224
U 1278   3226 3227 3249 3250 3251 3252 3254 3255
U 1280   3280 3281 3282 3283 3284 3285 3287 3288
U 1288   3289 3291 3292 3315 3316 3317 3319 3320
U 1290   3343 3446 3447 3448 3449 3450 3451 3452
U 1298   3453 3454 3461 3463 3465 3466 3470 3471
U 12A0   3480 3484 3485 3494 3495 3496 3525 3526
U 12A8   3528 3529 2127: 3552 3554 3555 3567 3571
U 12B0   3572 3581 3582 3583 3584 3588 3619 3620
U 12B8   3622 3623 3646 3647 3649 3650 3654 3717
U 12C0   3718 3719 3720 3721 3723 3724 3729 3732
U 12C8   3755 3756 3760 3761 3763 3764 3766 3804
U 12D0   3805 3842 3843 3844 3846 3847 3848 3947
U 12D8   3948 3949 3950 3951 3952 4079 4080 4081
U 12E0   4082 4083 4084 4123 4124 4132 4134 4135
U 12E8   4140 4247 4248 4249 4250 4252 4254 4260
U 12F0   4261 4262 4279 4280 4281 4282 4283 4284
U 12F8   4285 4290 4334 4335 4336 4381 4382 4383
U 1300   4385 4424 4425 4426 4427 4469 4470 4471
U 1308   4472 4480 4481 4484 4485 4486 4487 4490
U 1310   4491 4492 4493 4494 4501 4502 4503 4511
U 1318   4512 4514 4516 4517 4529 4530 4531 4532
U 1320   4533 4535 4545 4546 4547 4548 4549 4550
U 1328   4551 4552 4553 4554 4555 4556 4557 4558
U 1330   4559 4560 4561 4562 4600 4601 4641 4642
U 1338   4645 4646 4647 4648 4658 4659 4660 4665
U 1340   4667 4668 4674 4675 4676 4677 4682 4683
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 0761
ESKAR45.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:23          Page  15

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4684 4685 4686 4691 4807 4808 4829 4830
U 1350  4831 4841 4842 4843 4844 4849 4883 4885
U 1358  4891 4926 4990 4992 4993 4994 4996 4998
U 1360  5022 5023 5047 5048 5054 5055 5056 5057
U 1368  5058 5059 5060 5061 5063 5077 5078 5079
U 1370  5088 5089 5091 5092 5096 5097 5129 5130
U 1378  5136 5172 5173 5235 5237 5261 5262 5268
U 1380  5269 5270 5271 5272 5273 5274 5275 5277
U 1388  5279 5282 5300
U 1390 - 13EF Unused
U 13F0  2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8  2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR45	923
Used	923
Remaining	

Total microwords used in memory U: 923
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0763
ESKAR45.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:23     Page  15

; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR45.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR46.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 0764

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2173 Diagnostic Specific Subroutines
; 2174   Select Controller
; 2212   64K OR 256K CHIPS
; 2266   START TEST
; 2475   SELECT LOWER CONTROLLER
; 2523   SELECT UPPER CONTROLLER
; 2571   Set Diagnostic Mode Routine
; 2600   SBI Unjam Routine
; 2649   RESET SUBTEST NUMBER
; 2671   Initialize the SBI
; 2698   Disable Cache Routine
; 2720   Check for Interrupt Routine
; 2767   CHECK UTRAP
; 2799   Set Trap Mask
; 2827   FIND MS780-E MEMORY
; 2969   Generate IO Address
; 2994   Set Starting Address
; 3027   ENABLE NEXT MS780-E
; 3079   Wait Loop Routine
; 3110   CHECK SBI.ERR
; 3141   Single CPU State Routine
; 3158   Single Bus Cycle Routine
; 3175 TEST 1AD SINGLE STATE SBI INTERFACE REFERENCES
; 3493 TEST 1AE SINGLE TIME STATE ARRAY/SBI INTERFACE REF.
; 4070 TEST 1AF SINGLE BUS STATE SBI INTERFACE REFERENCES
; 4391 TEST 1B0 SINGLE BUS STATE ARRAY/SBI INTERFACE REF.
; 4968 TEST 1B1 RESERVED
; 4973 TEST 1B2 RESERVED
; 4978 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR46
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
;1841  .PAGE
;1842
;1843  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844  .TOC "   Micro Trap Handler"
;1845 ;++
;1846 ; FUNCTIONAL DESCRIPTION:
;1847 ;
;1848 ; This routine is responsible for 'catching' micro-traps
;1849 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850 ; contains the Trap-Expected Mask. If the specified bit is set in
;1851 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852 ; an error call is made.
;1853 ;
;1854 ; INPUT PARAMETERS:
;1855 ;
;1856 ; R[0F] - Expected Trap Mask
;1857 ;  Trap Code Function
;1858 ;  ---------      --------
;1859 ;       0  System Init
;1860 ;       1  Data Unaligned
;1861 ;       2  Cross Page
;1862 ;       3  TB Modify
;1863 ;       4  TB Protection
;1864 ;       6  TB Miss
;1865 ;       7  TB Parity
;1866 ;       8  Cache Parity
;1867 ;       9  Reserved Floating Operand
;1868 ;       C  Read Data Substitute
;1869 ;       D  Time out
;1870 ;       F  Control Store Parity Error
;1871 ;      10  Odd Address
;1872 ;
;1873 ; OUTPUT PARAMETERS:
;1874 ;
;1875 ; R[0F] - Mask bit is cleared.
;1876 ;
;1877 ; DATA: Micro PC of Micro Trap
;1878 ;  Trap Code (See Above)
;1879 ;  Fault Status Register
;1880 ;  SBI Error Register
;1881 ;  Timeout Address Register
;1882 ;  Maintenance Register
;1883 ;  Cache Parity Register
;1884 ;  TB Error Register 1
;1885 ;  TB Error Register 0
;1886 ;--
U 1100, 0000.003C.0180.0800.0084.7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000.003C.0580.0800.0084.7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000.003C.0980.0800.0084.7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000.003C.0D80.0800.0084.7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000.003C.1180.0800.0084.7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000.003C.D580.0800.0084.7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000.003C.D980.0800.0084.7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000.003C.5D80.0800.0084.7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000.003C.0180.0800.0084.7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000,003C,8580,0800,0084,7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000,003C,8980,0800,0084,7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000,003C,6580,0800,0084,7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000,003C,6180,0800,0084,7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1900
U 1001, 0000,003C,81F0,2C00,0000,104B  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 104B, 0C00,003C,01E0,0800,0000,104F  ;1902  q_d,d_q   ; Setup to put it back on stack
U 104F, 0C00,003C,81E0,3C00,0000,105B  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 105B, 0000,003C,01C0,0A78,0000,105F  ;1904  q_r[0f]   ; Get the Expected Trap Mask
        ;1905  alu_q.andnot.mask,
U 105F, 0001,2024,0180,0800,0010,1097  ;1906   clk.ubcc ; Was trap expected?
U 1097, 0000,013C,0180,0800,0000,1002  ;1907  z?
        ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
        ;1909       alu_q.and.mask, ; ...
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1910       return0  ; ...
U 1003, 0018,0038,1D80,09E8,0000,104C  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 104C, 0000,003D,D980,0800,0084,71C8  ;1912 =0 setrcf[.9]  ; Set output count
U 104D, 0000,003C,49F0,2C00,0000,1109  ;1913  q_id[tber0]  ; Output all the error registers
U 1109, 0001,203C,4DF0,2DB0,0000,11A8  ;1914  rc[6]_q,q_id[tber1] ; ...
U 11A8, 0001,203C,65F0,2DB8,0000,11A9  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 11A9, 0001,203C,6DF0,2DD8,0000,11AA  ;1916  rc[0b]_q,q_id[fault] ; ...
U 11AA, 0001,203C,75F0,2DE0,0000,11AB  ;1917  rc[0c]_q,q_id[maint] ; ...
U 11AB, 0001,203C,79F0,2DC8,0000,11AC  ;1918  rc[9]_q,q_id[parity] ; ...
U 11AC, 0001,203C,69F0,2DC0,0000,11AD  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 11AD, 0001,203C,81F0,2DD0,0000,1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
        ;1921 1000: ERROR1[.C],  ;
U 1000, 0001,203D,8580,09F0,0084,70BC  ;1922   rc[0e]_q ; Report the error
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "   Newtest Routine"
                ;1926 ;**
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to     0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 11AE, 0000.003C.65F8.0800.0084.71AF  ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 11AF, 0818.0038.8D80.0800.0000.11B0  ;1955  d_k[.1f]  ; D=1F
U 11B0, 0D00.003C.0180.0800.0000.11B1  ;1956  d_dal.sc  ; D=001F0000
U 11B1, 0000.003C.3D80.3C00.0000.11B2  ;1957  id[psl]_d  ; psl = 001F0000
U 11B2, 0818.0038.0580.0800.0000.11B3  ;1958  d_k[.1]  ; Clear the CES register
U 11B3, 0D00.003C.0180.0800.0000.11B4  ;1959  d_dal.sc  ; D = 00010000
U 11B4, 0F00.003C.3180.3C00.0000.11B5  ;1960  id[ces]_d,d_0  ; ces = 00010000
U 11B5, 0000.003C.5D80.3C00.0000.11B6  ;1961  id[acc.cs]_d  ; acc.cs = 0
U 11B6, 0000.003C.3980.3C00.0000.11B7  ;1962  id[sir]_d  ; sir = 0
U 11B7, 0000.003C.2980.3C00.0000.11B8  ;1963  id[clk.cs]_d  ; clk.cs = 0
U 11B8, 0000.003C.4980.3C00.0000.1058  ;1964  id[tber0]_d  ; tber0 = 0
U 1058, 0000.003D.0180.0800.0000.11E9  ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1966  intrpt.strobe,  ; Strobe interrupts
U 1059, 0818.0038.C180.0800.4000.11B9  ;1967      d_k[.ffff]  ; Setup to clear SBI error register and
U 11B9, 0801.0028.6580.3C00.0000.11BA  ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 11BA, 0F00.003C.6D80.3C00.0000.11BB  ;1969  id[fault]_d,d_0  ; ...
U 11BB, 0000.003C.6D80.3C00.0000.11BC  ;1970  id[fault]_d  ; fault = 0
U 11BC, 0000.003C.7580.3C00.0000.11BD  ;1971  id[maint]_d  ; MAINT = 0
U 11BD, 0000.003C.6580.3C00.0000.11BE  ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 11BE, 0000.003C.7180.3C00.0000.11BF  ;1973  id[comp]_d  ; comp reg = 0
U 11BF, 0000.003C.E580.3C00.0000.11C0  ;1974  ID[39]_d  ; Clear code for subroutine START.TEST
U 11C0, 0001.003C.0180.09F0.0000.11C1  ;1975  rc[0e]_d  ;
U 11C1, 0001.003C.0180.0AF8.0000.11C2  ;1976  r[0f]_d  ; Clear expected trap mask
U 11C2, 0001.003C.0180.09F8.0000.105C  ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,11EF  ;1978 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1980  .PAGE
            ;1981  .TOC "   Error Loop Routine"
            ;1982 ;**
            ;1983 ; FUNCTIONAL DESCRIPTION:
            ;1984 ;
            ;1985 ; This routine is called with the "ERLOOP" macro. The
            ;1986 ; routine calls the Micro Monitor to setup an error loop.
            ;1987 ;
            ;1988 ; CALLING CONSTRAINT:
            ;1989 ;
            ;1990 ; =0
            ;1991 ;
            ;1992 ; SIDE EFFECTS:
            ;1993 ;
            ;1994 ; D Reg - Destroyed
            ;1995 ;--
U 11C3, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 11C4, 0810,0038,0180,0978,0000,11C5  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 11C5, 0819,0034,4D80,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;++
            ;2025 ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 10BC, 0000,003D,0180,0800,0000,11C8  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10BD, 0000,003C,0180,0800,0000,11C4  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "   Error 2 Routine"
            ;2045 ;++
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 11C6. 0810,0038,0180,0978,0000,11C7  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 11C7. 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A. 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "   ERROR 2 WITH PARAMETER"
            ;2071 ;++
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 10BE, 0000,003D,0180,0800,0000,11C8  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10BF, 0000,003C,0180,0800,0000,11C6  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
            ;2088  .PAGE
            ;2089  .TOC "   Micro-Monitor Call"
            ;2090 ;*
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;**
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2111 13F0: nop
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2112 13F1: nop
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2113 13F2: nop
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2114 13F3: nop
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2115 13F4: nop
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2116 13F5: nop
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2117 13F6: nop
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2118 13F7: nop
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2119 13F8: nop
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2120 13F9: nop
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2121 13FA: nop
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2122 13FB: nop
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2123 13FC: nop
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2124 13FD: nop
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2125 13FE: nop
U 13FF, 0000,003C,0180,0800,0000,12AA  ;2126 13FF: nop
U 12AA, 0000,003C,0180,0800,0000,11C8  ;2127 12AA: nop ; This location is permanently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;++
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 11C8, 0010,0038,01C0,0978,0000,11C9  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 11C9, 0019,2034,4DC0,0800,0000,11CA  ;2150  q_q.and.k[.ff00] ; ...
U 11CA, 0019,2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,returni ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "   Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 11CB, 0810,0038,4D80,0978,0000,11CC  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 11CC, 0019,4016,4D80,09F8,0000,0001  ;2171       word,return1 ; (Subtest number is negative)
```

```
            ;2172  .PAGE
            ;2173  .TOC " Diagnostic Specific Subroutines"
            ;2174  .TOC "   Select Controller"
            ;2175 ;++
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine is used to put the memory system into the
            ;2179 ; specified configuration with respect to controller select.
            ;2180 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2181 ; a RETURN2 is made.
            ;2182 ;
            ;2183 ; CALLING CONSTRAINT:
            ;2184 ;
            ;2185 ; =00
            ;2186 ;
            ;2187 ; INPUTS:
            ;2188 ;
            ;2189 ; d - Contains value to write to bits<2:0> of Register A
            ;2190 ; rc[0e] - Address of Register A
            ;2191 ;
            ;2192 ; SIDE EFFECTS:
            ;2193 ;
            ;2194 ; sc,d,va are destroyed
            ;2195 ;--
            ;2196 SELECT.CNTRLR:
U 11CD, 0010,0038,0180,0970,0284,71CE  ;2197  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11CE, 0801,001C,0180,0800,0000,11CF  ;2198  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11CF, 0000,003C,0180,A800,0000,11D0  ;2199  cache.p_d[long]  ; Write the configuration Register
U 11D0, 0818,0038,5D80,0800,0000,11D1  ;2200  d_k[.7]  ; Get Misconfiguration bits
U 11D1, 0000,003C,61F8,0800,0084,71D2  ;2201  sc_k[.F],q_0  ; ...
U 11D2, 0D00,003C,0180,0800,0000,11D3  ;2202  d_del.sc  ; ... D = x38000
U 11D3, 0000,003C,01E0,0800,0000,11D4  ;2203  q_d  ; Save mask
U 11D4, 0000,003C,0180,C800,0000,11D5  ;2204  d[long]_cache.p  ; Read CNFG A Reg and
            ;2205  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11D5, 001D,2034,0180,0800,0010,11D6  ;2206  clk.ubcc ; ...
U 11D6, 0000,013C,0180,0800,0000,10C0  ;2207  z?  ; Branch if no
U 10C0, 0000,003E,0180,0800,0000,0001  ;2208 =0 RETURN1  ; Bad configuration
U 10C1, 0000,003E,0180,0800,0000,0002  ;2209  RETURN2  ; Configuration ok
            ;2210
```

```
          ;2211  .PAGE
          ;2212  .TOC "   64K OR 256K CHIPS"
          ;2213 ;*************************************************************
          ;2214 ;++
          ;2215 ;
          ;2216 ; Functional Description:
          ;2217 ; ----------------------
          ;2218 ; This subroutine is used to find the type of chips
          ;2219 ; on the MS780-E/H arrays.  The RETURN modes are as
          ;2220 ; follows:
          ;2221 ;
          ;2222 ; RETURN1 = Error. Mixed 64K and 256K arrays
          ;2223 ;
          ;2224 ; RETURN2 = 64K chips on the array
          ;2225 ;
          ;2226 ; RETURN3 = 256K chips on the array
          ;2227 ;
          ;2228 ; Calling Constraint: =00
          ;2229 ; ------------------
          ;2230 ;
          ;2231 ;
          ;2232 ; Inputs:
          ;2233 ; ------
          ;2234 ;
          ;2235 ; RC[0E] must hold the I/O base address of the MS780-E/H
          ;2236 ; currently under test
          ;2237 ;
          ;2238 ; Outputs:
          ;2239 ; -------
          ;2240 ;
          ;2241 ; NO specific outputs. (See RETURN modes above)
          ;2242 ;
          ;2243 ;
          ;2244 ; Side Effects:
          ;2245 ; ------------
          ;2246 ;
          ;2247 ; The contents of the following registers will be lost:
          ;2248 ;
          ;2249 ; D
          ;2250 ;
          ;2251 ;--
          ;2252 ;*************************************************************
          ;2253 64K.OR.256K:
U 11D7, 0010,0038,0180,0970,0200,11D8  ;2254  VA_RC[0E]    ; Point to CNFG Register A
U 11D8, 0000,003C,0180,C800,0000,11D9  ;2255  D[LONG]_CACHE.P    ; Read the register
U 11D9, 0200,003C,0180,0800,0000,11DA  ;2256  D_D.RIGHT2    ; Shift received contents two bits
          ;2257       ; ...to the right
U 11DA, 0600,003C,0180,0800,0000,11DB  ;2258  D_D.RIGHT    ; Shift D one more time
U 11DB, 0000,193C,0180,0800,0000,100C  ;2259  D1-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2260 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2261  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2262  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2263  RETURN1     ; Error: missing arrays
          ;2264
```

```
;2265  .PAGE
;2266  .TOC "   START TEST"
;2267 ;*************************************************************
;2268 ;**
;2269 ;
;2270 ; Functional Description:
;2271 ; ---------------------
;2272 ;
;2273 ;         This subroutine will be called by all tests that  check  the
;2274 ;    controllers,  or  those  tests  that  have  to switch between the
;2275 ;    controllers when executing.  Its  main  functions  are:  select
;2276 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2277 ;    test that called it, call out an error  if  neither  one  of  the
;2278 ;    controllers can be configured properly, if all the controllers on
;2279 ;    a MS780-E/H were configured and tested, it should select the next
;2280 ;    MS780-E/H on the SBI and repeat the above sequence.
;2281 ;    A test making use of this  subroutine  should  be  configured  as
;2282 ;    follows:
;2283 ;
;2284 ;
;2285 ;       Begin TEST.XX
;2286 ;        :
;2287 ;        :   Call NEWTST
;2288 ;        :   Call FIND.MS780E
;2289 ;        :   IF No MS780-E's on the SBI
;2290 ;        :     THEN
;2291 ;        :         Skip to End of TEST.XX
;2292 ;        :     ELSE
;2293 ;        :
;2294 ;        : RESTART.TEST.XX:
;2295 ;        :
;2296 ;        :   Call START TEST
;2297 ;        :   IF Return1 from START.TEST
;2298 ;        :     THEN
;2299 ;        :         Skip to End of TEST.XX
;2300 ;        :     ELSE.
;2301 ;        :
;2302 ;        ~
;2303 ;
;2304 ;        :
;2305 ;        :   Body of TEST.XX
;2306 ;        :
;2307 ;        ~
;2308 ;
;2309 ;        :   Jump RESTART.TEST.XX
;2310 ;        :
;2311 ;       End TEST.XX
;2312 ;
;2313 ;
;2314 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2315 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2316 ;    should be passed when the subroutine is called a second, third or
;2317 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2318 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2319 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2320 ;    Register 39 can be interpreted as follows:
;2321 ;
;2322 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2323 ;               called the first time.  When this code is encountered
;2324 ;               this subroutine will try to select the lower controller.
;2325 ;               If the lower controller is misconfigured, the pointer in
;2326 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2327 ;               is re-entered.
;2328 ;               If, however there is no misconfiguration, the value in
;2329 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2330 ;               calling test.
;2331 ;
;2332 ;     b01   -   This value signifies that an attempt at configuring the
;2333 ;               lower controller failed, and the subroutine will attempt
;2334 ;               to select the upper controller.
;2335 ;               If the upper controller is also misconfigured execution
;2336 ;               stops with a call to ERROR1.
;2337 ;               If there is no misconfiguration on this controller, then
;2338 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2339 ;               made to the calling test.
;2340 ;
;2341 ;     b10   -   This value signifies that the lower controller was selected
;2342 ;               previously and the test was executed once.  If the
;2343 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2344 ;               39 is loaded with b11 and an attempt is made to select the
;2345 ;               upper controller.
;2346 ;               If there is a misconfiguration on this controller, this
;2347 ;               subroutine is just re-entered.
;2348 ;               Otherwise, a Return2 will be made to the calling test.
;2349 ;
;2350 ;     b11   -   This code identifies the cases in which the test has
;2351 ;               been executed at least once (i.e., at least one of the
;2352 ;               controllers on the MS780-E unit under test could be
;2353 ;               configured properly).  When this code is detected the
;2354 ;               subroutine clears ID Reg 39 and makes a call to
;2355 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2356 ;               tested then the subroutine executes a Return1.
;2357 ;               Otherwise the subroutine is re-entered.
;2358 ;
;2359 ;
;2360 ; Calling Constraint: =00
;2361 ; ------------------
;2362 ;
;2363 ;
;2364 ; Inputs:
;2365 ; ------
;2366 ;
;2367 ; ID Register 39 must be cleared by NEWTST
;2368 ;
;2369 ;
;2370 ; Outputs:
;2371 ; -------
;2372 ;
;2373 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2374 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2375 ;         to be tested
            ;2376 ;
            ;2377 ; Side Effects:
            ;2378 ; ------------
            ;2379 ;
            ;2380 ; Contents of the following regiters will be destroyed:
            ;2381 ;
            ;2382 ; D, Q
            ;2383 ;
            ;2384 ;--
            ;2385 ;*************************************************************
            ;2386 =0
            ;2387 START.TEST:
U 10C2, 0000,003D,0180,0800,0000,11EA  ;2388  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 10C3, 0000,003C,0180,0800,0000,10C4  ;2389  NOP    ; ...tests that have more than one
            ;2390 =0      ; ...subtest.)
U 10C4, 0000,003D,0180,0800,0000,11C3  ;2391  ERLOOP    ; Set up the error loop for the
            ;2392       ; ...eventuality that the MS780-E/H
            ;2393       ; ...currently under test has no
            ;2394       ; ...Controllers
            ;2395 RE.ENTRY:    ; Re entry point for this routine
U 10C5, 0000,003C,E5F0,2C00,0000,11DC  ;2396  Q_ID[39]   ; Get the code
U 11DC, 0C00,003C,0180,0800,0000,11DD  ;2397  D_Q    ; Put it in the D Register
U 11DD, 0000,193C,0180,0800,0000,101C  ;2398  DI-0?    ; Branch on the code
U 101C, 0000,003C,0180,0800,0000,1008  ;2399 =1100 J/STRT.CASE00   ; Code is 00
U 101D, 0000,003C,0180,0800,0000,1010  ;2400  J/STRT.CASE01   ; Code is 01
U 101E, 0000,003C,0180,0800,0000,11E2  ;2401  J/STRT.CASE10   ; Code is 10
U 101F, 0000,003C,0180,0800,0000,1017  ;2402  J/STRT.CASE11   ; Code is 11
            ;2403 ;
            ;2404 ;
            ;2405 ; At this point the code in ID[39] was 00
            ;2406 ;
            ;2407 ;
            ;2408 =00
            ;2409 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1020  ;2410  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2411  J/STRT.LOW.MIS,   ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,11DE  ;2412   D_K[.1]   ; Set up the code for re entry to this
            ;2413       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2414  D_K[.2]    ; Set up the code for a next call to
            ;2415       ; ...START.TEST indicating that the
            ;2416       ; ...Lower Controller was configured
            ;2417       ; ...( 10 )
            ;2418  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2419   RETURN2   ; Let the test know that the Lower
            ;2420       ; ...has been configured
            ;2421 STRT.LOW.MIS:
            ;2422  ID[39]_D,    ; Save code in ID[39] signifing that
U 11DE, 0000,003C,E580,3C00,0000,10C5  ;2423   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2424       ; ...and re enter this subroutine
            ;2425
            ;2426 ;
            ;2427 ;
            ;2428 ; At this point the code is 01
            ;2429 ;
```

```
            ;2430 ;
            ;2431 =00
            ;2432 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1024  ;2433  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2434  J/STRT.UPR.MIS,   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,11DF  ;2435   D_0   ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2436  D_K[.3]    ; Upper controller was configured
            ;2437       ; ...thus the code must be changed
            ;2438       ; ...accordingly ( 11 )
            ;2439  ID[39]_D,   ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2440   RETURN2   ; Let the test know that the Upper
            ;2441       ; ...Controller has been configured
            ;2442 STRT.UPR.MIS:
U 11DF, 0000,003C,E580,3C00,0000,11E0  ;2443  ID[39]_D   ; Save the Code ( 00 )
U 11E0, 0018,0038,0D80,09E8,0000,11E1  ;2444  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 11E1, 0000,003D,0980,0800,0084,70BC  ;2445  ERROR1[.2]   ; Flag the error.
            ;2446 ;
            ;2447 ;
            ;2448 ; At this point the code is 10
            ;2449 ;
            ;2450 ;
            ;2451 STRT.CASE10:
U 11E2, 0818,0038,0D80,0800,0000,1014  ;2452  D_K[.3]    ; Set the code to 11
            ;2453
            ;2454
            ;2455 =00 ID[39]_D,   ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1024  ;2456   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,10C5  ;2457  J/RE.ENTRY   ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2458  RETURN2    ; Upper Controller configured
            ;2459       ; ...let the test know it
            ;2460 ;
            ;2461 ;
            ;2462 ; At this point the code is 11
            ;2463 ;
            ;2464 ;
            ;2465 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1018  ;2466  D_0    ; Clear the code
            ;2467 =00 ID[39]_D,   ; Save the code
U 1018, 0000,003D,E580,3C00,0000,123B  ;2468   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2469       ; ...on the SBI
U 1019, 0000,003C,0180,0800,0000,10C5  ;2470  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2471       ; ...attempt to configure the cntrllrs
U 101A, 0000,003E,0180,0800,0000,0001  ;2472  RETURN1    ; No more MS780-E/Hs found
U 101B, 0000,003C,0180,0800,0000,1020  ;2473  NOP
```

```
            ;2474  .PAGE
            ;2475  .TOC "   SELECT LOWER CONTROLLER"
            ;2476 ;*************************************************************
            ;2477 ;++
            ;2478 ;
            ;2479 ; Functional Description:
            ;2480 ; ----------------------
            ;2481 ;
            ;2482 ; This subroutine can be called to select the lower
            ;2483 ; Controller on a MS780-E/H.
            ;2484 ; If the Lower controller is misconfigured then a
            ;2485 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2486 ;
            ;2487 ; Calling Constraint: =00
            ;2488 ; ------------------
            ;2489 ;
            ;2490 ;
            ;2491 ; Inputs:
            ;2492 ; ------
            ;2493 ;
            ;2494 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2495 ;
            ;2496 ; Outputs:
            ;2497 ; -------
            ;2498 ;
            ;2499 ; RC[0D] will have the address of CNFG Register C
            ;2500 ;        ( 2000x008 )
            ;2501 ;
            ;2502 ; Side Effects:
            ;2503 ; ------------
            ;2504 ;
            ;2505 ; Contents of the following registers will lost:
            ;2506 ;
            ;2507 ; D
            ;2508 ;
            ;2509 ; The Lower controller on the MS780-E/H will be configured
            ;2510 ; when the subroutine is exitted with a RETURN2
            ;2511 ;--
            ;2512 ;*************************************************************
            ;2513 =00
            ;2514 SELECT.LOWER:
            ;2515  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1020, 0818,0039,1980,0800,0000,11CD  ;2516   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000,003E,0180,0800,0000,0001  ;2517  RETURN1    ; Lower Controller Misconfigured
U 1022, 0810,0038,0180,0970,0000,1023  ;2518  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;2519  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1023, 0019,0016,0180,09E8,0000,0002  ;2520   RETURN2   ; Let caller know that the controller
            ;2521       ; ...was configured properly
```

```
;2522  .PAGE
;2523  .TOC "    SELECT UPPER CONTROLLER"
;2524 ;*************************************************************
;2525 ;++
;2526 ;
;2527 ; Functional Description:
;2528 ; ----------------------
;2529 ;
;2530 ; This subroutine can be called to select the upper
;2531 ; Controller on a MS780-E/H.
;2532 ; If the Upper controller is misconfigured then a
;2533 ; RETURN1 will be made.  Otherwise a RETURN2
;2534 ;
;2535 ; Calling Constraint: =00
;2536 ; ------------------
;2537 ;
;2538 ;
;2539 ; Inputs:
;2540 ; ------
;2541 ;
;2542 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2543 ;
;2544 ; Outputs:
;2545 ; -------
;2546 ;
;2547 ; RC[0D] will have the address of CNFG Register D
;2548 ;        ( 2000x010 )
;2549 ;
;2550 ; Side Effects:
;2551 ; ------------
;2552 ;
;2553 ; Contents of the following registers will lost:
;2554 ;
;2555 ; D
;2556 ;
;2557 ; The Upper controller on the MS780-E/H will be configured
;2558 ; when the subroutine is exitted with a RETURN2
;2559 ;--
;2560 ;*************************************************************
;2561 =00
;2562 SELECT.UPPER:
;2563  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,0980,0800,0000,11CD  ;2564   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2565  RETURN1    ; Upper Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2566  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2567  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 1027, 0019,0016,8580,09E8,0000,0002  ;2568   RETURN2   ; Let caller know that the controller
;2569       ; ...was configured properly
```

```
            ;2570  .PAGE
            ;2571  .TOC "   Set Diagnostic Mode Routine"
            ;2572 ;++
            ;2573 ; FUNCTIONAL DESCRIPTION:
            ;2574 ;
            ;2575 ; This routine is used to set the specified diagnostic mode
            ;2576 ; in Register B. Other bits in the register remain unchanged.
            ;2577 ;
            ;2578 ; CALLING CONSTRAINT:
            ;2579 ;
            ;2580 ; =0
            ;2581 ;
            ;2582 ; INPUTS:
            ;2583 ;
            ;2584 ; d - Contains the mode to set in bits<1:0>
            ;2585 ; rc[0e] - Contains base address of controller
            ;2586 ;
            ;2587 ; SIDE EFFECTS:
            ;2588 ;
            ;2589 ; d,va,sc are destroyed
            ;2590 ;--
            ;2591 SET_DIAG_MODE:
U 11E3, 0010,0038,D9F8,0970,0284,71E4  ;2592  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11E4, 0D00,003C,0180,0803,0000,11E5  ;2593  va_va+4,d_dal.sc ; Shift specified mode into position
U 11E5, 0000,003C,01E0,0800,0000,11E6  ;2594  q_d   ; Save the diagnostic mode bits
U 11E6, 0000,003C,0180,C800,0000,11E7  ;2595  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11E7, 081D,0030,0180,0800,0000,11E8  ;2596  d_d.or.q  ; Set the diagnostic mode bits
U 11E8, 0000,003E,0180,A800,0000,0001  ;2597  cache.p_d[long],return1 ; Set the specified mode and return
            ;2598
```

```
            ;2599  .PAGE
            ;2600  .TOC "   SBI Unjam Routine"
            ;2601 ;++
            ;2602 ; FUNCTIONAL DESCRIPTION:
            ;2603 ;
            ;2604 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2605 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2606 ; and finally HOLD for the last 16 cycles.
            ;2607 ;
            ;2608 ; CALLING CONSTRAINT:
            ;2609 ;
            ;2610 ; =0
            ;2611 ;
            ;2612 ; SIDE EFFECTS:
            ;2613 ;
            ;2614 ; D Reg destroyed
            ;2615 ;--
            ;2616   ;
            ;2617   ; assert hold for 16 cycles
            ;2618   ;
            ;2619 SBI.UNJAM:
            ;2620  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11E9, 0818,0038,6580,8000,0010,10C6  ;2621      clk.ubcc  ; set micro cc's for next cycle
            ;2622 =0
            ;2623 SBI.UNJAM.1:
            ;2624  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2625       clk.ubcc,z?, ; ...
U 10C6, 0819,0100,0580,8000,0010,10C6  ;2626      j/sbi.unjam.1 ; test for completion
            ;2627    ;
            ;2628    ; assert hold and unjam for 16 cycles
            ;2629    ;
            ;2630  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10C7, 0818,0038,6580,8800,0010,10C8  ;2631      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2632 =0
            ;2633 SBI.UNJAM.2:
            ;2634  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2635       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10C8, 0819,0100,0580,8800,0010,10C8  ;2636      z?,j/sbi.unjam.2 ; ...
            ;2637    ;
            ;2638    ; assert hold for 16 cycles
            ;2639    ;
            ;2640  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10C9, 0818,0038,6580,8000,0010,10CA  ;2641      clk.ubcc  ; and set cc's for next cycle
            ;2642 =0
            ;2643 SBI.UNJAM.3:
            ;2644  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2645       clk.ubcc,z?, ; ...
U 10CA, 0819,0100,0580,8000,0010,10CA  ;2646      j/sbi.unjam.3 ; ...
U 10CB, 0000,003E,0180,0800,0000,0001  ;2647  return1   ; exit
```

```
            ;2648  .PAGE
            ;2649  .TOC "   RESET SUBTEST NUMBER"
            ;2650 ;++
            ;2651 ; FUNCTIONAL DESCRIPTION:
            ;2652 ;
            ;2653 ; Since some  of the tests will have to be restarted when two
            ;2654 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2655 ; the subtest counter to zero ( only for thoes tests that have
            ;2656 ; more than one subtest). This subroutine may also be necessary
            ;2657 ; when a test is being loop on.
            ;2658 ;
            ;2659 ; CALLING CONSTRAINT:
            ;2660 ;
            ;2661 ; =0
            ;2662 ; SIDE EFFECTS:
            ;2663 ;
            ;2664 ; D is destroyed
            ;2665 ;
            ;2666 ;--
            ;2667 RESET.SUBTST:
            ;2668  RC[0F]_0,   ; Reset subtest number
U 11EA, 0003,003E,0180,09F8,0000,0001  ;2669   RETURN1   ; ...and return
```

```
            ;2670  .PAGE
            ;2671  .TOC "   Initialize the SBI"
            ;2672 ;++
            ;2673 ; FUNCTIONAL DESCRIPTION:
            ;2674 ;
            ;2675 ; This routine performs the following functions:
            ;2676 ;
            ;2677 ;  Executes an SBI Unjam
            ;2678 ;  Clears the SBI Fault Latch
            ;2679 ;  Clears the SBI Error Register
            ;2680 ;
            ;2681 ; CALLING CONSTRAINT:
            ;2682 ;
            ;2683 ; =0
            ;2684 ;
            ;2685 ; SIDE EFFECTS:
            ;2686 ;
            ;2687 ; D & Q Register destroyed
            ;2688 ;--
            ;2689 =0
            ;2690 SBI.INIT:
U 10CC, 0000.003D.0180.0800.0000.11E9  ;2691  call[sbi.unjam]  ; Unjam the SBI
U 10CD, 0818.0038.C180.0800.0000.11EB  ;2692  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11EB, 0801.0028.6580.3C00.0000.11EC  ;2693  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11EC, 0F00.003C.6D80.3C00.0000.11ED  ;2694  id[fault]_d,d_0
U 11ED, 0000.003C.6D80.3C00.0000.11EE  ;2695  id[fault]_d
U 11EE, 0000.003E.6580.3C00.0000.0001  ;2696  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2697  .PAGE
            ;2698  .TOC "   Disable Cache Routine"
            ;2699 ;**
            ;2700 ; FUNCTIONAL DESCRIPTION:
            ;2701 ;
            ;2702 ; This routine disables cache hits by setting bits <16:15>
            ;2703 ; in the maintenance register.
            ;2704 ;
            ;2705 ; CALLING SEQUENCE:
            ;2706 ;
            ;2707 ; =0
            ;2708 ;
            ;2709 ; SIDE EFFECTS:
            ;2710 ;
            ;2711 ; D & Q Registers destroyed
            ;2712 ;--
            ;2713 DISABLE.CACHE:
U 11EF, 0818,0038,4580,0800,0000,11F0  ;2714  d_k[.8000]  ; ... and seed constant
U 11F0, 0500,003C,0180,0800,0000,11F1  ;2715  d_d.left  ; Generate final constant
U 11F1, 0819,0030,4580,0800,0000,11F2  ;2716  d_d.or.k[.8000]  ; ... = 00018000
U 11F2, 0000,003E,7580,3C00,0000,0001  ;2717  id[maint]_d,return1 ; Disable cache and return
            ;2718
```

```
            ;2719  .PAGE
            ;2720  .TOC "   Check for Interrupt Routine"
            ;2721 ;++
            ;2722 ; FUNCTIONAL DESCRIPTION:
            ;2723 ;
            ;2724 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2725 ; If an interrupt is active, the following registers are setup:
            ;2726 ;  RC[8] - Gets the VECTOR register
            ;2727 ;  RC[7] - Gets the SBI ERROR register
            ;2728 ;  RC[6] - Gets the FAULT register
            ;2729 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2730 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2731 ;
            ;2732 ; CALLING CONSTRAINT:
            ;2733 ;
            ;2734 ; =00
            ;2735 ;
            ;2736 ; SIDE EFFECTS:
            ;2737 ;
            ;2738 ; D & Q are destroyed
            ;2739 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2740 ;--
            ;2741 CHECK_INTERRUPT:
            ;2742   ;
            ;2743   ; First, enable the fault and RDS/CRD interrupts
            ;2744   ;
U 11F3. 0818.0038.4580.0800.0000.11F4  ;2745  d_k[.8000]  ; Get data to set interrupt enable
            ;2746  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 11F4. 0500.003C.6580.3C00.0000.11F5  ;2747       d_d.left
U 11F5. 0100.003C.0180.0800.0000.11F6  ;2748  d_d.left2  ; D = 40000
U 11F6. 0000.003C.6D80.3C00.0000.11F7  ;2749  id[fault]_d  ; Set fault interrupt enable
            ;2750  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 11F7. 0F03.003C.0180.09A8.4000.11F8  ;2751       rc[5]_0  ; and clear interrupt occurred flag
U 11F8. 0000.003C.3D80.3C00.0000.11F9  ;2752  id[psl]_d  ; Clear the PSL
U 11F9. 0000.003C.6580.3C00.0000.11FA  ;2753  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 11FA. 0000.003C.6D80.3C00.0000.11FB  ;2754  id[fault]_d  ; Clear FAULT Interrupt Enable
U 11FB. 0000.0E3C.0180.0800.0000.1004  ;2755  int?   ; Branch if external interrupt is active
U 1004. 0000.003E.0180.0800.0000.0001  ;2756 =100 return1   ; Exit with no interrupt
U 1005. 0000.003C.0180.0800.0000.1007  ;2757  j/check.int.1  ; Interrupt
U 1006. 0000.003E.0180.0800.0000.0001  ;2758  return1   ; No interrupt
            ;2759 =111
            ;2760 CHECK.INT.1:
U 1007. 0000.003C.35F0.2C00.0000.11FC  ;2761  q_id[vector]  ; Get ID Vector Register
U 11FC. 0001.203C.65F0.2DC0.0000.11FD  ;2762  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 11FD. 0001.203C.6DF0.2DB8.0000.11FE  ;2763  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 11FE. 0001.203C.0180.09B0.0000.11FF  ;2764  rc[6]_q   ; Save falut reg
U 11FF. 0018.003A.0580.09A8.0000.0002  ;2765  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2766  .PAGE
            ;2767  .TOC "   CHECK UTRAP"
            ;2768 ;++
            ;2769 ;FUNCTIONAL DESCRIPTION:
            ;2770 ;
            ;2771 ; This subroutine is used to check wether a Utrap occured.
            ;2772 ; It takes the contents of the Save Utrap flags register
            ;2773 ; R[0E], and compares them to the contrnts of the Utrap
            ;2774 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2775 ; according to the results of the test  (i.e., contents of
            ;2776 ; these registers are equal or not).
            ;2777 ; CALLING CONSTRAINT:
            ;2778 ;
            ;2779 ; =00
            ;2780 ;
            ;2781 ; INPUTS:
            ;2782 ;
            ;2783 ; R[0E]- Saved Utrap Mask
            ;2784 ; R[0F]- Expected Utrap Mask
            ;2785 ;
            ;2786 ;--
            ;2787 CHECK_UTRAP:
U 1200. 0800,003C,0180,0A70,0000,1201  ;2788  D_R[0E]    ; Reg D is loaded with the trap mask
U 1201. 0001,003C,0180,09C0,0000,1202  ;2789  RC[08]_D   ; Save expected Utrap flag for error logout
U 1202. 0800,003C,0180,0A78,0000,1203  ;2790  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1203. 0001,003C,0180,09B8,0000,1204  ;2791  RC[07]_D   ; Save in RC[07]
            ;2792  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1204. 000D,0020,0180,0A70,0010,1205  ;2793         CLK.UBCC
U 1205. 0003,013C,0180,0AF8,0000,10CE  ;2794  Z?,R[0F]_0   ; Branch if no traps
U 10CE. 0000,003E,0180,0800,0000,0002  ;2795 =0 RETURN2    ; Utrap occured
U 10CF. 0000,003E,0180,0800,0000,0001  ;2796  RETURN1    ; No Utrap  return
            ;2797 ;
```

```
            ;2798  .PAGE
            ;2799  .TOC "   Set Trap Mask"
            ;2800 ;**
            ;2801 ; FUNCTIONAL DESCRIPTION:
            ;2802 ;
            ;2803 ; This routine is used to set a bit in the Micro Trap Mask
            ;2804 ; R[0F].
            ;2805 ;
            ;2806 ; CALLING CONSTRAINT:
            ;2807 ;
            ;2808 ; =0
            ;2809 ;
            ;2810 ; INPUTS:
            ;2811 ;
            ;2812 ; SC - Contains bit number to set.
            ;2813 ;
            ;2814 ; OUTPUTS:
            ;2815 ;
            ;2816 ; rc[0E] - Contains the current trap mask
            ;2817 ;
            ;2818 ; SIDE EFFECTS:
            ;2819 ;
            ;2820 ; d regsiter is destroyed
            ;2821 ;--
            ;2822 SET_TRAP_MASK:
U 1206, 0800,003C,0180,0A78,0000,1207  ;2823  d_r[0f]
U 1207, 0801,001C,0180,0AF8,0000,1208  ;2824  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1208, 0001,003E,0180,0AF0,0000,0001  ;2825  r[0E]_d,return1   ; Save mask and return
```

```
            ;2826  .PAGE
            ;2827  .TOC "   FIND MS780-E MEMORY"
            ;2828 ;**
            ;2829 ; FUNCTIONAL DESCRIPTION:
            ;2830 ;
            ;2831 ;          The diagnostic is designed to handle up to 4 (four)
            ;2832 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2833 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2834 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2835 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2836 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2837 ;     first MS780-E found will have its starting address set to 0
            ;2838 ;     (zero).
            ;2839 ;     All the other MS780-E/H's found will have their starting address
            ;2840 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2841 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2842 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2843 ;     current MS780-E unit under test.
            ;2844 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2845 ;     set to maximum.
            ;2846 ;
            ;2847 ; CALLING CONSTRAINT:
            ;2848 ;
            ;2849 ; =00
            ;2850 ;
            ;2851 ; OUTPUTS:
            ;2852 ;
            ;2853 ; rc[0e] - Contains address of Configuration Register
            ;2854 ; r[0]  - Contains TR level
            ;2855 ;
            ;2856 ; SIDE EFFECTS:
            ;2857 ;
            ;2858 ; D register is destroyed.
            ;2859 ;--
            ;2860 =0
            ;2861 FIND.MS780E:
U 10D0, 0000.003D.0180.0800.0000.10CC  ;2862  call[sbi.init]  ; Make sure SBI is enabled
U 10D1, 0003.003C.0180.0988.0000.1209  ;2863  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1209, 0818.0038.1980.0800.0000.120A  ;2864  d_k[zero]  ; Clear MS780-E/H counter
U 120A, 0000.003C.F980.3C00.0000.120B  ;2865  id[3e]_d  ; ...
U 120B, 0018.0038.0580.0A80.0000.120C  ;2866  r[0]_k[.1]  ; Init TR counter
U 120C, 0F03.003C.0180.09F0.0000.10D2  ;2867  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2868      ; ...CNFG A Reg
            ;2869 =0
            ;2870 FIND.MEM.LOOP:
U 10D2, 0800.003D.0180.0A00.0000.1232  ;2871  d_r[0],call[generate.io]; Generate address of TR level
U 10D3, 0001.003C.0180.09F0.0200.10D4  ;2872  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10D4, 0000.003D.8980.0800.0084.7206  ;2873 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2874  d[long]_cache.p, ; Read the specified tr level
U 10D5, 0000.003C.0180.C970.0000.120D  ;2875   ic_rc[0e]  ; ...
U 120D, 0000.003C.0180.0A78.0010.120E  ;2876  alu_r[0f],clk.ubcc ; Check for no response
U 120E, 0001.013C.0180.0880.0082.10D6  ;2877  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10D6, 0000.003C.0180.0800.0000.120F  ;2878 =0 j/check.adpt.code ; Check for a memory
U 10D7, 0000.003C.0180.0800.0000.122E  ;2879  j/find.mem.lp1  ; Try next tr
            ;2880 CHECK.ADPT.CODE:
```

```
U 120F, 0000,003C,1D80,0800,1404,7210  ;2881  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 1210, 0000,163C,0180,0800,0000,1028  ;2882  state7-4?  ; Branch on the adapter type
U 1028, 0000,003C,8980,0800,0084,7211  ;2883 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1029, 0000,003C,8980,0800,0084,7212  ;2884  j/ms780.c,sc_k[.d] ; MS780-C
U 102A, 0000,003C,0180,0800,0000,122E  ;2885  j/find.mem.lpl  ; Not a memory
U 102B, 0000,003C,0180,0800,0000,122E  ;2886  j/find.mem.lpl  ; Not a memory
U 102C, 0000,003C,6580,0800,0084,7217  ;2887  j/ma780.max,sc_k[.10] ; MA780
U 102D, 0000,003C,0180,0800,0000,122E  ;2888  j/find.mem.lpl  ; Not a memory
U 102E, 0010,0038,0180,09F0,0000,121B  ;2889  rc[0e]_ic,j/found.ms780e ; MS780-E
U 102F, 0010,0038,0180,09F0,0000,121B  ;2890  rc[0e]_ic,j/found.ms780e ; MS780-E
            ;2891   ;
            ;2892   ; Found an MS780-A or -C. Set the starting address
            ;2893   ; to maximum.
            ;2894   ;
U 1211, 0818,0038,5D80,0800,0000,1213  ;2895 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1212, 0818,0038,0580,0800,0000,1213  ;2896 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2897 MS780.COMMON:
U 1213, 0819,0030,7180,0800,0000,1214  ;2898  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1214, 0000,003C,01F8,0803,0080,D215  ;2899  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1215, 0D00,003C,0180,0800,0000,1216  ;2900  d_dal.sc  ; Put data in bits<29:14>
            ;2901  cache.p_d[long], ; Set the starting address to maximum
U 1216, 0000,003C,0180,A800,0000,122E  ;2902       j/find.mem.lpl ; and continue TR scan
            ;2903   ;
            ;2904   ; Found an MA780. Set the starting address to maximum
            ;2905   ;
            ;2906 MA780.MAX:
U 1217, 0818,0038,39F8,0803,0000,1218  ;2907  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1218, 0D00,003C,0180,0803,0000,1219  ;2908  d_dal.sc,va_va+4 ; Put in proper position
U 1219, 0000,003C,0180,0803,0000,121A  ;2909  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2910  cache.p_d[long], ; Set starting address to maximum
U 121A, 0000,003C,0180,A800,0000,122E  ;2911       j/find.mem.lpl
            ;2912 ;
            ;2913 ; Found an MS780E
            ;2914 ;
            ;2915 FOUND.MS780E:
U 121B, 0000,003C,F9F0,2C00,0000,121C  ;2916  q_id[3e]  ; Set up to see how many MS780-E's
            ;2917  alu_q.xor.k[.3], ; Are there Maximum units?
U 121C, 0019,2020,0D80,0800,0010,121D  ;2918   clk.ubcc ; Check condition codes
U 121D, 0000,013C,0180,0800,0000,10D8  ;2919  z?    ; Are we at maximum units for system
U 10D8, 0000,003C,0180,0800,0000,121E  ;2920 =0 J/ms780e.tst  ; No go find how many units ther are
U 10D9, 0818,0038,C180,0800,0000,10DA  ;2921  d_k[.ffff]  ; Yes set address to maximum
U 10DA, 0000,003D,0180,0800,0000,1236  ;2922 =0 call[set.start.addr] ; .....
U 10DB, 0000,003C,0180,0800,0000,122E  ;2923  j/find.mem.lpl  ; Go check to see if we are done
            ;2924 ;
            ;2925 MS780E.TST:
            ;2926  alu_rc[01],  ; Check "Found MS780E Flag"
U 121E, 0010,0038,0180,0908,0010,121F  ;2927   clk.ubcc ; Check condition codes
U 121F, 0810,0138,0180,0970,0000,10DC  ;2928  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2929 =0 j/not.first.ms780e, ; No
U 10DC, 0818,0038,C180,0800,0000,10E0  ;2930   d_k[.ffff] ; Set starting address
U 10DD, 0001,003C,0180,0988,0000,1220  ;2931  rc[01]_d  ; Yes put base address in rc[01]
U 1220, 0818,0038,7980,0800,0000,1221  ;2932  d_k[.30]  ; Set up SC to point to first unit
U 1221, 0019,0014,F580,0800,0082,1222  ;2933  alu_d+k[.a],sc_alu ; ...
U 1222, 0810,0038,0180,0908,0000,1223  ;2934  d_rc[01]  ; Put base address of unit in D
U 1223, 0000,003C,0180,3400,0000,1224  ;2935  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1224. 0F00.003C.0180.0800.0000.10DE  ;2936  d_0   ; Prepare to set starting address to Zero
U 10DE. 0000.003D.0180.0800.0000.1236  ;2937 =0 call[set.start.addr] ; Go do it
          ;2938  j/find.mem.lp1,  ; Check to see if we are done
U 10DF. 0003.003C.0180.09E8.0000.122E  ;2939   rc[0d]_0 ; ....
          ;2940 =0
          ;2941 NOT.FIRST.MS780E:
U 10E0. 0000.003D.0180.0800.0000.1236  ;2942  call[set.start.addr] ; Go set starting address to maximum
U 10E1. 0000.003C.F9F0.2C00.0000.1225  ;2943  q_id[3e]  ; Get the number of MS780-Es found
U 1225. 0819.2014.0580.0800.0000.1226  ;2944  d_q+k[.1]  ; Increment this counter
U 1226. 0000.003C.F980.3C00.0000.1227  ;2945  id[3e]_d  ; Save the counter
U 1227. 0018.0038.11C0.0800.0000.1228  ;2946  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2947     ; ...were the I/O base addresses will be stored
U 1228. 081D.2000.7980.0800.0000.1229  ;2948  d_q-d,k[.30]  ; ... adjust pointer
U 1229. 0819.0014.7980.0800.0000.122A  ;2949  d_d+k[.30]  ; Point the SC to the ID register
U 122A. 0000.003C.F580.0800.0000.122B  ;2950  k[.a]   ; ...to receive I/O base address
U 122B. 0019.0014.F580.0800.0082.122C  ;2951  alu_d+k[.a],sc_alu ; ....
U 122C. 0810.0038.0180.0970.0000.122D  ;2952  d_rc[0e]  ; Get base address to d
U 122D. 0000.003C.0180.3400.0000.122E  ;2953  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2954   ;
          ;2955   ; Try next tr
          ;2956   ;
          ;2957 FIND.MEM.LP1:
U 122E. 0818.0014.0580.0A80.0000.122F  ;2958  r[0]_la+k[.1],d_alu ; Increment TR level
          ;2959  alu_d.and.k[.f],
U 122F. 0019.0034.6180.0800.0010.1230  ;2960      clk.ubcc  ; Check if all done
U 1230. 0000.013C.0180.0800.0000.10E2  ;2961  z?   ; Branch if yes
U 10E2. 0000.003C.0180.0800.0000.10D2  ;2962 =0 j/find.mem.loop  ; Try next TR
U 10E3. 0810.0038.0180.0908.0010.1231  ;2963  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1231. 0000.013C.0180.0800.0000.10E4  ;2964  z?   ; Check condition codes
U 10E4. 0001.003E.0180.09F0.0000.0002  ;2965 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10E5. 0000.003E.0180.0800.0000.0001  ;2966  return1   ; No MS780E found
          ;2967
```

```
            ;2968   .PAGE
            ;2969   .TOC "   Generate IO Address"
            ;2970  ;++
            ;2971  ; FUNCTIONAL DESCRIPTION:
            ;2972  ;
            ;2973  ; This routine is used to generate an SBI Configuration Register
            ;2974  ; address from a TR level.
            ;2975  ;
            ;2976  ; CALLING CONSTRAINT:
            ;2977  ;
            ;2978  ; =0
            ;2979  ;
            ;2980  ; INPUTS:
            ;2981  ;
            ;2982  ; D - Contains TR level
            ;2983  ;
            ;2984  ; OUTPUTS:
            ;2985  ;
            ;2986  ; D - Contains SBI IO address
            ;2987  ;--
            ;2988 GENERATE.IO:
U 1232, 0000,003C,89F8,0800,0084,7233   ;2989  sc_k[.d],q_0   : Setup to shift TR level 13 bits
U 1233, 0D00,003C,8D80,0800,0084,7234   ;2990  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 1234, 0000,003C,0980,0800,0084,B235   ;2991  sc_sc-k[.2]   ; insert bit 29
U 1235, 0801,001E,0180,0800,0000,0001   ;2992  d_d.ornot.mask,return1 ; and return
```

```
            ;2993  .PAGE
            ;2994  .TOC "   Set Starting Address"
            ;2995 ;++
            ;2996 ; FUNCTIONAL DESCRIPTION:
            ;2997 ;
            ;2998 ; This routine is used to set the starting address of the
            ;2999 ; memory to the specified value.
            ;3000 ;
            ;3001 ; CALLING CONSTRAINT:
            ;3002 ;
            ;3003 ; =0
            ;3004 ;
            ;3005 ; INPUTS:
            ;3006 ;
            ;3007 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3008 ;     (B Register Image divided by 2**16)
            ;3009 ; rc[0e] - Contains Address of Register A
            ;3010 ;
            ;3011 ; OUTPUTS:
            ;3012 ;
            ;3013 ; d - Contains aligned starting address (B Register Image)
            ;3014 ;
            ;3015 ; SIDE EFFECTS:
            ;3016 ;
            ;3017 ; d,q,va, and sc are destroyed
            ;3018 ;--
            ;3019 SET.START.ADDR:
U 1236, 0010,0038,6580,0970,0284,7237  ;3020  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1237, 0000,003C,01F8,0803,0000,1238  ;3021  va_va+4,q_0  ; Set address to Register B
            ;3022  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1238, 0D00,003C,0980,0800,0084,B239  ;3023      sc_sc-k[.2]  ; Setup to insert bit 14
U 1239, 0801,001C,0180,0800,0000,123A  ;3024  d_d.ornot.mask  ; Insert Write Enable bit
U 123A, 0000,003E,0180,A800,0000,0001  ;3025  cache.p_d[long],return1 ; Set the starting address and return
```

```
            ;3026  .PAGE
            ;3027  .TOC "   ENABLE NEXT MS780-E"
            ;3028 ;++
            ;3029 ; FUNCTIONAL DESCRIPTION:
            ;3030 ;
            ;3031 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3032 ;    They will be tested  separately, according to the TR level at which
            ;3033 ;    they are located,  starting with the one  at the lowest level first.
            ;3034 ;          When  a test has finished  with the first unit, it will have to call
            ;3035 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3036 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3037 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3038 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3039 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3040 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3041 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3042 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3043 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3044 ;
            ;3045 ; CALLING CONSTRAINT:
            ;3046 ;
            ;3047 ;  =00
            ;3048 ;
            ;3049 ;--
            ;3050 ENABLE.NEXT.MS780E:
U 123B, 0000.003C.F9F0.2C00.0000.123C  ;3051  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3052  ALU_Q,   ; Check to see if it is zero
U 123C, 0001.203C.0180.0800.0010.123D  ;3053     CLK.UBCC  ; Check Condition Codes
U 123D, 0000.013C.0180.0800.0000.10E6  ;3054  Z?   ; ...for zero
U 10E6, 0000.003C.0180.0800.0000.123E  ;3055 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10E7, 0000.003E.0180.0800.0000.0002  ;3056  RETURN2   ; Zero No more units to test
            ;3057 ENABLE.NEXT:
U 123E, 0818.0038.C180.0800.0000.10E8  ;3058  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10E8, 0000.003D.0180.0800.0000.1236  ;3059 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10E9, 0818.0038.7980.0800.0000.123F  ;3060  D_K[.30]  ; Prepare to point to the ID register holding
            ;3061     ; ...the I/O Base address of the next MS780-E
U 123F, 0819.0014.1180.0800.0000.1240  ;3062  D_D+K[.4]  ; ...
U 1240, 0000.003C.F580.0800.0000.1241  ;3063  K[.A]   ; ...
U 1241, 0819.0014.F580.0800.0000.1242  ;3064  D_D+K[.A]  ; ...
U 1242, 0000.003C.F9F0.2C00.0000.1243  ;3065  Q_ID[3E]  ; Get MS780-E Counter
            ;3066  D_D-Q,   ; D now holds the pointer to the ID register
U 1243, 081D.0000.0180.0800.0082.1244  ;3067    SC_ALU  ; ...
U 1244, 0000.003C.01F0.2400.0000.1245  ;3068  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1245, 0001.203C.0180.09F0.0000.1246  ;3069  RC[0E]_Q  ; Save it also in RC[0E]
U 1246, 0F03.003C.0180.09E8.0000.10EA  ;3070  RC[0D]_0,D_0  ; Set starting address to zero
U 10EA, 0000.003D.0180.0800.0000.1236  ;3071 =0 CALL[SET.START.ADDR] ; ....
U 10EB, 0F00.003C.F9F0.2C00.0000.1247  ;3072  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1247, 081D.2008.0180.0800.0000.1248  ;3073  D_Q-D-1   ; Subtract 1 from total
U 1248, 0000.003C.F980.3C00.0000.10EC  ;3074  ID[3E]_D  ; Adjust the pointer
U 10EC, 0000.003D.0180.0800.0000.11EA  ;3075 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10ED, 0000.003E.0180.0800.0000.0001  ;3076  RETURN1   ; Tell caller another unit was found
            ;3077
```

```
            ;3078  .PAGE
            ;3079  .TOC "   Wait Loop Routine"
            ;3080 ;**
            ;3081 ; FUNCTIONAL DESCRIPTION:
            ;3082 ;
            ;3083 ; This routine is used to wait the specified number of machine cycles.
            ;3084 ; This routine is typically called to wait for an SBI write to
            ;3085 ; I/O space to complete.
            ;3086 ;
            ;3087 ; CALLING CONSTRAINT:
            ;3088 ;
            ;3089 ; =0
            ;3090 ;
            ;3091 ; INPUTS:
            ;3092 ;
            ;3093 ; d - Contains number of cycles to wait
            ;3094 ; alu.z condition code must not be set
            ;3095 ;
            ;3096 ; SIDE EFFECTS:
            ;3097 ;
            ;3098 ; d is destroyed
            ;3099 ;--
            ;3100 WAIT.LOOP:
U 1249, 0018,0038,6180,0800,0010,10EE  ;3101  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;3102      ; ...is not set
            ;3103 =0
            ;3104 W.LOOP:
            ;3105  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10EE, 0819,0100,0580,0800,0010,10EE  ;3106       j/W.LOOP
U 10EF, 0000,003E,0180,0800,0000,0001  ;3107  return1    ; exit
            ;3108
```

```
            ;3109  .PAGE
            ;3110  .TOC "   CHECK SBI.ERR"
            ;3111 ;**
            ;3112 ; FUNCTIONAL DESCRIPTION:
            ;3113 ;
            ;3114 ; This subroutine can be used to see whether the SBI.ERR Register
            ;3115 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;3116 ; routine will have to pass to this subroutine in the SC register
            ;3117 ; the number of the error bit to checked.
            ;3118 ;
            ;3119 ; CALLING CONSTRAINT:
            ;3120 ;
            ;3121 ; =00
            ;3122 ;
            ;3123 ; INPUTS:
            ;3124 ;
            ;3125 ; SC with the number of the error bit to be checked
            ;3126 ;
            ;3127 ; OUTPUTS:
            ;3128 ;
            ;3129 ; RC[08]-Contains the SBI.ERROR Register
            ;3130 ;--
            ;3131 CHECK_SBI_ERR:
U 124A. 0000.003C.65F0.2C00.0000.124B  ;3132  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 124B. 0001.203C.0180.09C0.0000.124C  ;3133  RC[08]_Q  ; Save for error logout
            ;3134  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 124C. 0001.2024.0180.0800.0010.124D  ;3135   CLK.UBCC ; ...
U 124D. 0000.013C.0180.0800.0000.10F0  ;3136  Z?   ; Check condition codes
U 10F0. 0000.003E.0180.0800.0000.0002  ;3137 =0 RETURN2   ; Bit is set
U 10F1. 0000.003E.0180.0800.0000.0001  ;3138  RETURN1   ; Bit is not set
            ;3139
```

```
            ;3140  .PAGE
            ;3141  .TOC "   Single CPU State Routine"
            ;3142 ;+
            ;3143 ; FUNCTIONAL DESCRIPTION:
            ;3144 ;
            ;3145 ; This routine is used to call the monitor to perform
            ;3146 ; single time states.
            ;3147 ;
            ;3148 ; EXPLICIT INPUTS:
            ;3149 ;
            ;3150 ; D<15:0> Reg contains number of time states to tick.
            ;3151 ;--
            ;3152 STEP:
U 124E, 0000,003C,65F8,0800,0084,724F  ;3153  sc_k[.10],q_0  ; Put step count in byte 2,3
U 124F, 0D00,003C,8980,0800,0000,1250  ;3154  d_dal.sc,k[.d)  ; ...
U 1250, 0819,0010,8980,0800,0000,105E  ;3155  d_d+k[.d]+1,j/callcon ; Enter routine code and call console
            ;3156
```

```
            ;3157  .PAGE
            ;3158  .TOC "    Single Bus Cycle Routine"
            ;3159 ;*
            ;3160 ; FUNCTIONAL DESCRIPTION:
            ;3161 ;
            ;3162 ; This routine is used to call the monitor to perform
            ;3163 ; single bus cycles.
            ;3164 ;
            ;3165 ; EXPLICIT INPUTS:
            ;3166 ;
            ;3167 ; D<15:0> Reg contains number of bus cycles to tick.
            ;3168 ;--
            ;3169 BUS:
U 1251, 0000,003C,65F8,0800,0084,7252  ;3170  sc_k[.10],q_0  ; Put step count in byte 2,3
U 1252, 0D00,003C,0180,0800,0000,1253  ;3171  d_dal.sc  ;...
U 1253, 0819,0030,2180,0800,0000,1254  ;3172  d_d.or.k[.14]  ; Enter routine code
U 1254, 0819,0014,0980,0800,0000,105E  ;3173  d_d+k[.2],j/callcon ; and call monitor
            ;3174
```

```
;3175  .PAGE "TEST 1AD SINGLE STATE SBI INTERFACE REFERENCES"
;3176 ;++
;3177 ; Functional Description:
;3178 ; ---------------------
;3179 ;
;3180 ; This test checks whether the SBI Interface can perfrom write/read
;3181 ; operations while the CPU is in Single Time State Mode.
;3182 ;
;3183 ;   Subtest 1
;3184 ;
;3185 ; The Data Latches are written with a test pattern while the CPU is being
;3186 ; single stepped by the LSI-11.  A normal read is performed to check data
;3187 ; integrity.
;3188 ;
;3189 ;   Subtest 2
;3190 ;
;3191 ; A normal write operation is made to the Data Latches.  A Read will
;3192 ; follow with the CPU being single stepped, and the data retreived will
;3193 ; be check for errors.
;3194 ;
;3195 ; ----------------
;3196 ;
;3197 ;
;3198 ; Error Logout:
;3199 ; ------------
;3200 ;
;3201 ; See individual error reports.
;3202 ;
;3203 ;
;3204 ;
;3205 ;+++++++
```

```
U 10F2, 0000,003D,0180,0800,0000,11AE  ;3230 T.23: NEWTST   ; Signify start of new test
U 10F3, 0000,003C,0180,0800,0000,1030  ;3231  NOP
U 1030, 0000,003D,0180,0800,0000,10D0  ;3232 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
          ;3233      ; (Implicit enable of SBI)
U 1031, 0000,003C,0180,0800,0000,1047  ;3234  J/T.23.EXIT  ; No memories
          ;3235  ;
          ;3236  ; This is the point (RESTART.TEST.23) where after testing the functions
          ;3237  ; the program will return to here if there is another MS780-E on the
          ;3238  ; system and repeat the sequence of instructions.
          ;3239  ;
          ;3240 RESTART.TEST.23:
U 1032, 0000,003C,D980,0800,0084,7033  ;3241  SC_K[.9]  ; SC gets x9 for mask of bit 9
U 1033, 0803,001C,0180,0800,0000,1255  ;3242  D_NOT.MASK  ; Load the value x200 into the D register
U 1255, 0001,003C,0180,09C8,0000,1256  ;3243  RC[09]_D  ; Save x200 in RC[09]
U 1256, 0010,0038,01C0,0970,0000,1257  ;3244  Q_RC[0E]  ; Move MS780-E base address to Q
U 1257, 0019,2014,6580,09D0,0000,10F4  ;3245  RC[0A]_Q+K[.10]  ; Point address to Data Latch 1
          ;3246 ;
          ;3247 ;================================================================================
          ;3248 ;
          ;3249 ; Subtest 1 Register Writes
          ;3250 ;
U 10F4, 0000,003D,0180,0800,0000,11CB  ;3251 =0 SUBTEST   ; Increment Subtest count
U 10F5, 0000,003C,0180,0800,0000,10F6  ;3252  NOP
U 10F6, 0000,003D,0180,0800,0000,11C3  ;3253 =0 ERLOOP   ; Loop on error from here
U 10F7, 0000,003C,0180,0800,0000,10F8  ;3254  NOP
U 10F8, 0000,003D,0180,0800,0000,10CC  ;3255 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10F9, 0010,0038,0180,0950,0200,1258  ;3256  VA_RC[0A]  ; Load VA with address of Data latch 1
U 1258, 0F00,003C,0180,0800,0000,1259  ;3257  D_0   ; Init contents of data latch 1
U 1259, 0000,003C,0180,A800,0000,125A  ;3258  CACHE.P_D[LONG]  ; to zero
U 125A, 0810,0038,0180,0948,0000,10FA  ;3259  D_RC[09]  ; Load D with the number of cycles to step
U 10FA, 0000,003D,0180,0800,0000,124E  ;3260 =0 CALL[STEP]  ; Start single stepping
          ;3261  ;
          ;3262  ; Following sequence is performed in single time state mode
          ;3263  ;
U 10FB, 081B,001C,1980,0800,0000,125B  ;3264  D_NOT.K[ZERO]  ; Setup to write all ones to data latch 1
U 125B, 0001,003C,0180,09E0,0000,125C  ;3265  RC[0C]_D  ; Save write data in RC[0C] in case error
U 125C, 0000,003C,0180,A800,0000,10FC  ;3266  CACHE.P_D[LONG]  ; Start the write
          ;3267 =0 CALL[WAIT.LOOP],
U 10FC, 0810,0039,0180,0948,0010,1249  ;3268       D_RC[09],CLK.UBCC ; Wait in single step mode incase timeout
          ;3269  ;
          ;3270  ; End of single time state mode
          ;3271  ;
U 10FD, 0000,003C,0180,0800,0000,1034  ;3272  NOP
U 1034, 0000,003D,8580,0800,0084,724A  ;3273 =00 CHECK.SBI.ERR[.C] ; Was there a timeout on the write ?
U 1035, 0000,003C,0180,0800,0000,1038  ;3274  J/T.23.S1.NOTMO  ; No, go check for any unexpected interrupts
```

```
            ;3275  .PAGE
            ;3276 ;
            ;3277 ;////////////////////////////////////////////////////////////////////////////
            ;3278 ;
            ;3279 ; Got a time out ( SBI Error Bit <12> ) on the write.  Call an error
            ;3280 ;
U 1036. 0000.003D.5D80.0800.0084.70BE  ;3281 =0 ERROR2[.7]
U 1037. 0000.003C.0180.0800.0000.1038  ;3282  NOP
            ;3283 ;
            ;3284 ;////////////////////////////////////////////////////////////////////////////
            ;3285 ;
            ;3286 ; ERROR LOGOUT:
            ;3287 ;
            ;3288 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3289 ;  CNFG C/D Address of MS780-E Currently under test
            ;3290 ;  Unused
            ;3291 ;  Unused
            ;3292 ;  Address Under Test
            ;3293 ;  Cycle Count for Single Stepping
            ;3294 ;  ID[sbi.err] Register
            ;3295 ;
            ;3296 ;////////////////////////////////////////////////////////////////////////////
            ;3297 ;
            ;3298  ;
            ;3299  ; Check for unexpected Interrupt.
            ;3300  ;
            ;3301 =00
            ;3302 T.23.S1.NOTMO:
U 1038. 0000.003D.0180.0800.0000.11F3  ;3303  CHECK.INTERRUPT  ; Any unexpected interrupts ?
U 1039. 0000.003C.0180.0800.0000.103B  ;3304  J/T.23.S1.NOINT  ; No interrupt
```

```
            ;3305   .PAGE
            ;3306 ;
            ;3307 ;////////////////////////////////////////////////////////////////////////////////
            ;3308 ;
            ;3309 ; Got an unexpected interrupt on the write.  Call an error
            ;3310 ;
U 103A, 0000,003D,F580,0800,0084,70BE  ;3311  ERROR2[.A]
            ;3312 ;
            ;3313 ;////////////////////////////////////////////////////////////////////////////////
            ;3314 ;
            ;3315 ; ERROR LOGOUT:
            ;3316 ;
            ;3317 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3318 ;  CNFG C/D Address of MS780-E Currently under test
            ;3319 ;  Unused
            ;3320 ;  Unused
            ;3321 ;  Address Under Test
            ;3322 ;  Cycle Count for Single Stepping
            ;3323 ;  ID[vector] Register
            ;3324 ;  ID[sbi.err] Register
            ;3325 ;  ID[fault] Register
            ;3326 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;3327 ;       0 = No Interrupt Occurred
            ;3328 ;
            ;3329 ;////////////////////////////////////////////////////////////////////////////////
            ;3330 ;
            ;3331  ;
            ;3332  ; Check the Data to see if it is correct
            ;3333  ;
            ;3334 T.23.S1.NOINT:
U 103B, 0010,0038,0180,0950,0200,125D  ;3335  VA_RC[0A]  ; Load VA with test address
U 125D, 0000,003C,0180,C800,0000,125E  ;3336  D[LONG]_CACHE.P  ; Read data latch 1
U 125E, 0001,003C,0180,09D8,0000,125F  ;3337  RC[0B]_D  ; Store data read in case of an error
U 125F, 0019,0014,0580,0800,0010,1260  ;3338  ALU_D+K[.1],CLK.UBCC ; Was latch written ok?
U 1260, 0000,013C,0180,0800,0000,10FE  ;3339  Z?   ; Branch if yes
```

```
            ;3340  .PAGE
            ;3341  ;
            ;3342  ;////////////////////////////////////////////////////////////////////////////
            ;3343  ;
            ;3344  ; Got the wrong data when reading the data back.  Call an error
            ;3345  ;
U 10FE, 0000,003D,D580,0800,0084,70BE  ;3346 =0 ERROR2[.6]
U 10FF, 0000,003C,0180,0800,0000,110A  ;3347  NOP
            ;3348  ;
            ;3349  ;////////////////////////////////////////////////////////////////////////////
            ;3350  ;
            ;3351  ; ERROR LOGOUT:
            ;3352  ;
            ;3353  ; DATA: I/O Base Address of MS780-E Currently under test
            ;3354  ;  CNFG C/D Address of MS780-E Currently under test
            ;3355  ;  Expected Memory Data
            ;3356  ;  Received Memory Data
            ;3357  ;  Address Under Test
            ;3358  ;  Cycle Count for Single Stepping
            ;3359  ;
            ;3360  ;////////////////////////////////////////////////////////////////////////////
            ;3361  ;
            ;3362  ;
            ;3363  ;============================================================================
            ;3364  ;
            ;3365  ; Subtest 2 Register Read
            ;3366  ;
U 110A, 0000,003D,0180,0800,0000,11CB  ;3367 =0 SUBTEST   ; Increment subtest counter
U 110B, 0000,003C,0180,0800,0000,1110  ;3368  NOP
U 1110, 0000,003D,0180,0800,0000,11C3  ;3369 =0 ERLOOP   ; Set the error loop
U 1111, 0000,003C,0180,0800,0000,1112  ;3370  NOP
U 1112, 0000,003D,0180,0800,0000,10CC  ;3371 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 1113, 0010,0038,0180,0950,0200,1261  ;3372  VA_RC[0A]  ; Load VA with the test address
U 1261, 081B,001C,1980,0800,0000,1262  ;3373  D_NOT.K[ZERO]
U 1262, 0000,003C,0180,A800,0000,1114  ;3374  CACHE.P_D[LONG]  ; Write one's
U 1114, 0000,003D,8980,0800,0084,7206  ;3375 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 1115, 0000,003C,0180,0800,0000,1263  ;3376  NOP
U 1263, 0810,0038,0180,0948,0000,1116  ;3377  D_RC[09]  ; D gets the number of cycles to single step
U 1116, 0000,003D,0180,0800,0000,124E  ;3378 =0 CALL[STEP]  ; Start single stepping
            ;3379  ;
            ;3380  ; Following sequence is performed in single time state mode
            ;3381  ;
U 1117, 0000,003C,0180,C800,0000,1264  ;3382  D[LONG]_CACHE.P  ; Start the read
U 1264, 0001,003C,0180,09D8,0000,1118  ;3383  RC[0B]_D  ; Save read data
            ;3384 =0 CALL[WAIT.LOOP],
U 1118, 0810,0039,0180,0948,0010,1249  ;3385      D_RC[09],CLK.UBCC ; Wait in single step mode incase timeout
            ;3386  ;
            ;3387  ; End of single time state mode
            ;3388  ;
U 1119, 0000,003C,0180,0800,0000,103C  ;3389  NOP
U 103C, 0000,003D,0180,0800,0000,1200  ;3390 =00 CHECK.UTRAP  ; Was there a Time Out micro trap ?
U 103D, 0000,003C,0180,0800,0000,1040  ;3391  J/T.23.S2.NOTMO  ; No, check for unexpected interrupts
```

```
            ;3392  .PAGE
            ;3393 ;
            ;3394 ;////////////////////////////////////////////////////////////////////////////////
            ;3395 ;
            ;3396 ; Got a time out micro trap on the read from the register.  Call an error.
            ;3397 ;
U 103E, 0000,003D,0180,0800,0084,70BE  ;3398  ERROR2[.8]
U 103F, 0000,003C,0180,0800,0000,1040  ;3399  NOP
            ;3400 ;
            ;3401 ;////////////////////////////////////////////////////////////////////////////////
            ;3402 ;
            ;3403 ; ERROR LOGOUT:
            ;3404 ;
            ;3405 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3406 ;  CNFG C/D Address of MS780-E Currently under test
            ;3407 ;  Expected Memory Data
            ;3408 ;  Received Memory Data
            ;3409 ;  Address Under Test
            ;3410 ;  Cycle Count for Single Stepping
            ;3411 ;  Expected Micro Trap Flags
            ;3412 ;  Received Micro Trap Flags
            ;3413 ;
            ;3414 ;////////////////////////////////////////////////////////////////////////////////
            ;3415 ;
            ;3416  ;
            ;3417  ; Check for any unexpected interrupts
            ;3418  ;
            ;3419 =00
            ;3420 T.23.S2.NOTMO:
U 1040, 0000,003D,0180,0800,0000,11F3  ;3421  CHECK.INTERRUPT  ; Check for interrupts
U 1041, 0000,003C,0180,0800,0000,1043  ;3422  J/T.23.S2.NOINT  ; No interrupt
```

```
            ;3423  .PAGE
            ;3424 ;
            ;3425 ;////////////////////////////////////////////////////////////////////////////////
            ;3426 ;
            ;3427 ; Got an unexpected interrupt on the read.  Call an error.
            ;3428 ;
U 1042, 0000,003D,F580,0800,0084,70BE  ;3429  ERROR2[.A]
            ;3430 ;
            ;3431 ;////////////////////////////////////////////////////////////////////////////////
            ;3432 ;
            ;3433 ; ERROR LOGOUT:
            ;3434 ;
            ;3435 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3436 ;  CNFG C/D Address of MS780-E Currently under test
            ;3437 ;  Unused
            ;3438 ;  Unused
            ;3439 ;  Address Under Test
            ;3440 ;  Cycle Count for Single Stepping
            ;3441 ;  ID[vector] Register
            ;3442 ;  ID[sbi.err] Register
            ;3443 ;  ID[fault] Register
            ;3444 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;3445 ;       0 = No Interrupt Occurred
            ;3446 ;
            ;3447 ;////////////////////////////////////////////////////////////////////////////////
            ;3448 ;
            ;3449  ;
            ;3450  ; check the data to insure that it is correct.
            ;3451  ;
            ;3452 T.23.S2.NOINT:
U 1043, 0810,0038,0180,0958,0000,1265  ;3453  D_RC[0B]  ; Move read data to D for validity check
U 1265, 0019,0014,0580,0800,0010,1266  ;3454  ALU_D+K[.1],CLK.UBCC ; Was latch written ok?
U 1266, 0000,013C,0180,0800,0000,111A  ;3455  Z?   ; Branch if data ok
```

```
            ;3456  .PAGE
            ;3457 ;
            ;3458 ;////////////////////////////////////////////////////////////////////////////
            ;3459 ;
            ;3460 ; Data read was not correct.  Call an error.
            ;3461 ;
U 111A, 0000,003D,D580,0800,0084,70BE  ;3462 =0 ERROR2[.6]
U 111B, 0000,003C,0180,0800,0000,1044  ;3463  NOP
            ;3464 ;
            ;3465 ;////////////////////////////////////////////////////////////////////////////
            ;3466 ;
            ;3467 ; ERROR LOGOUT:
            ;3468 ;
            ;3469 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3470 ;  CNFG C/D Address of MS780-E Currently under test
            ;3471 ;  Expected Memory Data
            ;3472 ;  Received Memory Data
            ;3473 ;  Address Under Test
            ;3474 ;  Cycle Count for Single Stepping
            ;3475 ;
            ;3476 ;////////////////////////////////////////////////////////////////////////////
            ;3477 ;
            ;3478  ;
            ;3479  ; At this point we have complete the bus functions under test.  The
            ;3480  ; next thing is to try enabling the next MS780-E and if there is indeed
            ;3481  ; another one then go to RESTART.TEST.23 and run test over, if not then
            ;3482  ; exit this test.
            ;3483  ;
U 1044, 0000,003D,0180,0800,0000,123B  ;3484 =00 CALL[ENABLE.NEXT.MS780E] ; Check and see if more MS780-E's ?
U 1045, 0000,003C,0180,0800,0000,1032  ;3485  J/RESTART.TEST.23  ; Yes, more to test, restart test
U 1046, 0000,003C,0180,0800,0000,1047  ;3486  NOP    ; No more to test, exit
            ;3487  ;
            ;3488  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;3489  ;
            ;3490 T.23.EXIT:
U 1047, 0000,003C,0180,0800,0000,111C  ;3491  NOP
            ;3492
```

```
;3493  .PAGE "TEST 1AE SINGLE TIME STATE ARRAY/SBI INTERFACE REF."
;3494 ;
;3495 ;*****************************************************************************
;3496 ;++
;3497 ; Functional Description:
;3498 ; ----------------------
;3499 ;
;3500 ; This test makes use of a routine executing in the monitor (on the
;3501 ; LSI-11) to single step the CPU.  Its main function is to verify that
;3502 ; array references are executed properly when the CPU is in the single
;3503 ; step mode.  No interrupts or uTraps are expected.  The following
;3504 ; functions are checked:
;3505 ;
;3506 ;    i.  Memory Write
;3507 ;   ii.  Memory Read
;3508 ;  iii.  Interlocked Write
;3509 ;   iv.  Interlocked Read
;3510 ;
;3511 ;   Subtest 1
;3512 ;
;3513 ; This subtest writes some data pattern to the array in Single CPU State.
;3514 ; The write is followed by a "normal" read operation to check the
;3515 ; integrity of the data.
;3516 ;
;3517 ;   Subtest 2
;3518 ;
;3519 ; This subtest initializes the array to some known data pattern and then
;3520 ; executes a read cycle in single CPU state.  The data returned is
;3521 ; checked for integrity against the original pattern.
;3522 ;
;3523 ;   Subtest 3
;3524 ;
;3525 ; This subtest is similar to Subtest 1, except that the memory cycle will
;3526 ; be an interlock write.  The array will be initialized with an Interlock
;3527 ; Read to avoid generating a Interlock Sequence Fault.
;3528 ;
;3529 ;   Subtest 4
;3530 ;
;3531 ; This subtest is similar to Subtest 2, except that the memory cycle
;3532 ; executed in Single CPU State will be an Interlock Read.
;3533 ;
;3534 ; -----------------
;3535 ;
;3536 ;
;3537 ; Error Logout:
;3538 ; ------------
;3539 ;
;3540 ; See individual error reports.
;3541 ;
;3542 ; ----------------------
;3543 ;
;3544 ;
;3545 ;=============================================================================
;3546 ;
;3547 ; Register Map:
```

```
            ;3548 ; ------------
            ;3549 ;
            ;3550 ; RA,RB  Description
            ;3551 ; -----  -----------
            ;3552 ;
            ;3553 ;      Not Used
            ;3554 ;
            ;3555 ; RC  Description
            ;3556 ; --  -----------
            ;3557 ;
            ;3558 ; E  I/O Base Address of MS780-E Currently Under Test
            ;3559 ;
            ;3560 ; C  Expected Memory Data
            ;3561 ;
            ;3562 ; B  Received Memory Data
            ;3563 ;
            ;3564 ; A  Address  Under Test
            ;3565 ;
            ;3566 ; 9  Cycle Count for Stepping CPU
            ;3567 ;
            ;3568 ;--
            ;3569 ;*****************************************************************************
            ;3570 =0
U 111C, 0000,003D,0180,0800,0000,11AE  ;3571 T.24: NEWTST
U 111D, 0000,003C,0180,0800,0000,1048  ;3572  NOP
U 1048, 0000,003D,0180,0800,0000,10D0  ;3573 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1049, 0000,003C,0180,0800,0000,1289  ;3574  J/T.24.EXIT  ; No memories, exit this test
U 104A, 0000,003C,0180,0800,0000,1050  ;3575  NOP
            ;3576 =
            ;3577  ;
            ;3578  ; This is the point where the program loops back to after executing one
            ;3579  ; pass of this test, if there are more MS780-E's to test on the system.
            ;3580  ;
            ;3581 =00
            ;3582 RESTART.TEST.24:
U 1050, 0000,003D,0180,0800,0000,10C2  ;3583  CALL[START.TEST] ; Configure the MS780-E
U 1051, 0000,003C,0180,0800,0000,1289  ;3584  J/T.24.EXIT  ; Couldn't configure or no more left
U 1052, 0000,003C,D980,0800,0084,7053  ;3585  SC_K[.9]  ; SC gets x9 for mask of bit 9
U 1053, 0803,001C,0180,0800,0000,1267  ;3586  D_NOT.MASK  ; Load the value x200 into the D register
U 1267, 0001,003C,0180,09C8,0000,1268  ;3587  RC[09]_D  ; Save in RC[09]
U 1268, 0018,0038,1980,09D0,0000,111E  ;3588  RC[0A]_K[ZERO]  ; Set-up RC[0A] with address 0 for test
            ;3589 ;
            ;3590 ;=============================================================================
            ;3591 ;
            ;3592 ; Subtest 1 Array Writes
            ;3593 ;
U 111E, 0000,003D,0180,0800,0000,11CB  ;3594 =0 SUBTEST   ; Set subtest counter
U 111F, 0000,003C,0180,0800,0000,1120  ;3595  NOP
U 1120, 0000,003D,0180,0800,0000,11C3  ;3596 =0 ERLOOP   ; Loop on error from here
U 1121, 0000,003C,0180,0800,0000,1122  ;3597  NOP
U 1122, 0000,003D,0180,0800,0000,10CC  ;3598 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1123, 0010,0038,0180,0950,0200,1269  ;3599  VA_RC[0A]  ; Point VA to correct address
U 1269, 0F00,003C,0180,0800,0000,126A  ;3600  D_0  ; Init the D reg to 0 to init memory location
U 126A, 0000,003C,0180,A800,0000,1124  ;3601  CACHE.P_D[LONG]  ; Initialize the memory location to zero
            ;3602 =0 D_RC[09],
```

```
U 1124, 0810,0039,0180,0948,0000,124E  ;3603      CALL[STEP]  ; Start single stepping
           ;3604  ;
           ;3605  ; Following sequence is performed in single time state mode
           ;3606  ;
U 1125, 081B,001C,1980,0800,0000,126B  ;3607  D_NOT.K[ZERO]  ; Setup to write all ones to array
U 126B, 0001,003C,0180,09E0,0000,126C  ;3608  RC[0C]_D  ; Save write data in RC[0C] in case error
U 126C, 0000,003C,0180,A800,0000,1126  ;3609  CACHE.P_D[LONG]  ; Start the write
           ;3610 =0 CALL[WAIT.LOOP],
U 1126, 0810,0039,0180,0948,0010,1249  ;3611      D_RC[09],CLK.UBCC ; Wait in single step mode incase timeout
           ;3612  ;
           ;3613  ; End of single time state mode
           ;3614  ;
U 1127, 0000,003C,0180,0800,0000,1054  ;3615  NOP
U 1054, 0000,003D,8580,0800,0084,724A  ;3616 =00 CHECK.SBI.ERR[.C] ; Was there a time out on the write ?
U 1055, 0000,003C,0180,0800,0000,1060  ;3617  J/T.24.S1.NOTMO  ; No timeout
```

```
            ;3618  .PAGE
            ;3619 ;
            ;3620 ;////////////////////////////////////////////////////////////////////////////////
            ;3621 ;
            ;3622 ; Got a time out ( SBI Error Bit <12> ) on the write to the memory location.
            ;3623 ; Call an error.
            ;3624 ;
U 1056. 0000.003D.5D80.0800.0084.70BE  ;3625  ERROR2[.7]
U 1057. 0000.003C.0180.0800.0000.1060  ;3626  NOP
            ;3627 ;
            ;3628 ;////////////////////////////////////////////////////////////////////////////////
            ;3629 ;
            ;3630 ; ERROR LOGOUT:
            ;3631 ;
            ;3632 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3633 ;  CNFG C/D Address of MS780-E Currently under test
            ;3634 ;  NOT USED
            ;3635 ;  NOT USED
            ;3636 ;  Address Under Test
            ;3637 ;  Cycle Count for Single Stepping
            ;3638 ;  ID[sbi.err] Register
            ;3639 ;
            ;3640 ;////////////////////////////////////////////////////////////////////////////////
            ;3641 ;
            ;3642  ;
            ;3643  ; Check for unexpected interrupts.
            ;3644  ;
            ;3645 =00
            ;3646 T.24.S1.NOTMO:
U 1060. 0000.003D.0180.0800.0000.11F3  ;3647  CHECK.INTERRUPT  ; Check for interrupts
U 1061. 0000.003C.0180.0800.0000.1063  ;3648  J/T.24.S1.NOINT  ; No interrupt
```

```
            ;3649  .PAGE
            ;3650 ;
            ;3651 ;////////////////////////////////////////////////////////////////////////////////
            ;3652 ;
            ;3653 ; Got an unexpected interrupt on the write.  Call an error.
            ;3654 ;
U 1062, 0000,003D,F580,0800,0084,70BE  ;3655  ERROR2[.A]
            ;3656 ;
            ;3657 ;////////////////////////////////////////////////////////////////////////////////
            ;3658 ;
            ;3659 ; ERROR LOGOUT:
            ;3660 ;
            ;3661 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3662 ;  CNFG C/D Address of MS780-E Currently under test
            ;3663 ;  NOT USED
            ;3664 ;  NOT USED
            ;3665 ;  Address Under Test
            ;3666 ;  Cycle Count for Single Stepping
            ;3667 ;  ID[vector] Register
            ;3668 ;  ID[sbi.err] Register
            ;3669 ;  ID[fault] Register
            ;3670 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;3671 ;       0 = No Interrupt Occurred
            ;3672 ;
            ;3673 ;////////////////////////////////////////////////////////////////////////////////
            ;3674 ;
            ;3675  ;
            ;3676  ; Check the Data to insure that it is correct.
            ;3677  ;
            ;3678 T.24.S1.NOINT:
U 1063, 0010,0038,0180,0950,0200,126D  ;3679  VA_RC[0A]  ; Load VA with the test address
U 126D, 0000,003C,0180,C800,0000,126E  ;3680  D[LONG]_CACHE.P  ; Read the array
U 126E, 0001,003C,0180,09D8,0000,126F  ;3681  RC[0B]_D  ; Store the data read in case of an error
U 126F, 0019,0014,0580,0800,0010,1270  ;3682  ALU_D+K[.1],CLK.UBCC ; Was data written ok?
U 1270, 0000,013C,0180,0800,0000,1128  ;3683  Z?   ; Branch if yes
```

```
            ;3684  .PAGE
            ;3685 ;
            ;3686 ;////////////////////////////////////////////////////////////////////////////
            ;3687 ;
            ;3688 ; Got the wrong data when reading the data back.  Call an error.
            ;3689 ;
U 1128. 0000.003D.D580.0800.0084.70BE  ;3690 =0 ERROR2[.6]
U 1129. 0000.003C.0180.0800.0000.112A  ;3691  NOP
            ;3692 ;
            ;3693 ;////////////////////////////////////////////////////////////////////////////
            ;3694 ;
            ;3695 ; ERROR LOGOUT:
            ;3696 ;
            ;3697 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3698 ;  CNFG C/D Address of MS780-E Currently under test
            ;3699 ;  Expected Memory Data
            ;3700 ;  Received Memory Data
            ;3701 ;  Address Under Test
            ;3702 ;  Cycle Count for Single Stepping
            ;3703 ;
            ;3704 ;////////////////////////////////////////////////////////////////////////////
            ;3705 ;
            ;3706 ;
            ;3707 ;============================================================================
            ;3708 ;
            ;3709 ; Subtest 2 Array Read
            ;3710 ;
U 112A. 0000.003D.0180.0800.0000.11CB  ;3711 =0 SUBTEST   ; Increment subtest counter
U 112B. 0000.003C.0180.0800.0000.112C  ;3712  NOP
U 112C. 0000.003D.0180.0800.0000.11C3  ;3713 =0 ERLOOP   ; Set the error loop
U 112D. 0000.003C.0180.0800.0000.112E  ;3714  NOP
U 112E. 0000.003D.0180.0800.0000.10CC  ;3715 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 112F. 0010.0038.0180.0950.0200.1271  ;3716  VA_RC[0A]  ; Load VA with test address
U 1271. 081B.001C.1980.0800.0000.1272  ;3717  D_NOT.K[ZERO]  ; Load D register with xFFFFFFFF and init
U 1272. 0001.003C.0180.09E0.0000.1273  ;3718  RC[0C]_D  ; Save write data in RC[0C] in case error
U 1273. 0000.003C.0180.A800.0000.1130  ;3719  CACHE.P_D[LONG]  ; Initialize the memory location
U 1130. 0000.003D.8980.0800.0084.7206  ;3720 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 1131. 0000.003C.0180.0800.0000.1132  ;3721  NOP
            ;3722 =0 D_RC[09],
U 1132. 0810.0039.0180.0948.0000.124E  ;3723       CALL[STEP]  ; Start single stepping
            ;3724  ;
            ;3725  ; Following sequence is performed in single time state mode
            ;3726  ;
U 1133. 0000.003C.0180.C800.0000.1274  ;3727  D[LONG]_CACHE.P  ; Start the read
U 1274. 0001.003C.0180.09D8.0000.1134  ;3728  RC[0B]_D  ; Save read data
            ;3729 =0 CALL[WAIT.LOOP],
U 1134. 0810.0039.0180.0948.0010.1249  ;3730       D_RC[09].CLK.UBCC ; Wait in single step mode incase timeout
            ;3731  ;
            ;3732  ; End of single time state mode
            ;3733  ;
U 1135. 0000.003C.0180.0800.0000.1064  ;3734  NOP
U 1064. 0000.003D.0180.0800.0000.1200  ;3735 =00 CHECK.UTRAP  ; Did we get a time out micro trap ?
U 1065. 0000.003C.0180.0800.0000.1068  ;3736  J/T.24.S2.NOTMO  ; No, go check for any unexpected interrupts
```

```
            ;3737  .PAGE
            ;3738 ;
            ;3739 ;////////////////////////////////////////////////////////////////////////////////
            ;3740 ;
            ;3741 ; Got a time out micro trap on the read.  Call an error.
            ;3742 ;
U 1066, 0000,003D,0180,0800,0084,70BE  ;3743  ERROR2[.8]
U 1067, 0000,003C,0180,0800,0000,1068  ;3744  NOP
            ;3745 ;
            ;3746 ;////////////////////////////////////////////////////////////////////////////////
            ;3747 ;
            ;3748 ; ERROR LOGOUT:
            ;3749 ;
            ;3750 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3751 ;  CNFG C/D Address of MS780-E Currently under test
            ;3752 ;  NOT USED
            ;3753 ;  NOT USED
            ;3754 ;  Address Under Test
            ;3755 ;  Cycle Count for Single Stepping
            ;3756 ;  Expected Micro Trap Flags
            ;3757 ;  Received Micro Trap Flags
            ;3758 ;
            ;3759 ;////////////////////////////////////////////////////////////////////////////////
            ;3760 ;
            ;3761  ;
            ;3762  ; Check for any unexpected interrupts.
            ;3763  ;
            ;3764 =00
            ;3765 T.24.S2.NOTMO:
U 1068, 0000,003D,0180,0800,0000,11F3  ;3766  CHECK.INTERRUPT  ; Check for interrupts
U 1069, 0000,003C,0180,0800,0000,106B  ;3767  J/T.24.S2.NOINT  ; No interrupt
```

```
            ;3768  .PAGE
            ;3769 ;
            ;3770 ;////////////////////////////////////////////////////////////////////////////////
            ;3771 ;
            ;3772 ; Got an unexpected interrupt on the read.  Call an error.
            ;3773 ;
U 106A, 0000.003D.F580.0800.0084.70BE  ;3774  ERROR2[.A]
            ;3775 ;
            ;3776 ;////////////////////////////////////////////////////////////////////////////////
            ;3777 ;
            ;3778 ; ERROR LOGOUT:
            ;3779 ;
            ;3780 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3781 ;  CNFG C/D Address of MS780-E Currently under test
            ;3782 ;  NOT USED
            ;3783 ;  NOT USED
            ;3784 ;  Address Under Test
            ;3785 ;  Cycle Count for Single Stepping
            ;3786 ;  ID[vector] Register
            ;3787 ;  ID[sbi.err] Register
            ;3788 ;  ID[fault] Register
            ;3789 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;3790 ;       0 = No Interrupt Occurred
            ;3791 ;
            ;3792 ;////////////////////////////////////////////////////////////////////////////////
            ;3793 ;
            ;3794  ;
            ;3795  ; Check the data to insure that it is correct.
            ;3796  ;
            ;3797 T.24.S2.NOINT:
U 106B, 0810.0038.0180.0958.0000.1275  ;3798  D_RC[0B]  ; Move data read to D register
U 1275, 0019.0014.0580.0800.0010.1276  ;3799  ALU_D+K[.1],CLK.UBCC ; Add 1 to result and should get 0
U 1276, 0000.013C.0180.0800.0000.1136  ;3800  Z?   ; Branch if data ok
```

```
            ;3801   .PAGE
            ;3802 ;
            ;3803 ;////////////////////////////////////////////////////////////////////////////
            ;3804 ;
            ;3805 ; Got the wrong data when reading the data back.  Call an error.
            ;3806 ;
U 1136, 0000,003D,D580,0800,0084,70BE  ;3807 =0 ERROR2[.6]
U 1137, 0000,003C,0180,0800,0000,1277  ;3808  NOP
            ;3809 ;
            ;3810 ;////////////////////////////////////////////////////////////////////////////
            ;3811 ;
            ;3812 ; ERROR LOGOUT:
            ;3813 ;
            ;3814 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3815 ;  CNFG C/D Address of MS780-E Currently under test
            ;3816 ;  Expected Memory Data
            ;3817 ;  Received Memory Data
            ;3818 ;  Address Under Test
            ;3819 ;  Cycle Count for Single Stepping
            ;3820 ;
            ;3821 ;////////////////////////////////////////////////////////////////////////////
            ;3822 ;
U 1277, 0000,003C,0180,0800,0000,1138  ;3823  NOP
            ;3824 ;
            ;3825 ;============================================================================
            ;3826 ;
            ;3827 ; Subtest 3 Interlock Write
            ;3828 ;
U 1138, 0000,003D,0180,0800,0000,11CB  ;3829 =0 SUBTEST   ; Update Subtest counter
U 1139, 0000,003C,0180,0800,0000,113A  ;3830  NOP
U 113A, 0000,003D,0180,0800,0000,11C3  ;3831 =0 ERLOOP   ; Set the error loop
U 113B, 0000,003C,0180,0800,0000,113C  ;3832  NOP
U 113C, 0000,003D,0180,0800,0000,10CC  ;3833 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 113D, 0F10,0038,0180,0950,0200,1278  ;3834  VA_RC[0A],D_0  ; Init contents of array
U 1278, 0000,003C,0180,A800,0000,1279  ;3835  CACHE.P_D[LONG]  ; to zero
U 1279, 0810,0038,0180,0948,0000,127A  ;3836  D_RC[09]  ; Move Cycle count to the D register
U 127A, 0819,0014,8D80,0800,0000,127B  ;3837  D_D+K[.1F]  ; Increase count to x21F to compensate for
            ;3838      ; doing the Lockread & lockwrite during step
U 127B, 0001,003C,0180,09E8,0000,113E  ;3839  RC[0D]_D  ; Save modified cycle count in RC[0D]
U 113E, 0000,003D,0180,0800,0000,124E  ;3840 =0 CALL[STEP]  ; Start single stepping
            ;3841  ;
            ;3842  ; Following sequence is performed in single time state mode
            ;3843  ;
U 113F, 0000,003C,0180,E800,0000,127C  ;3844  MCT/LOCKREAD.P  ; Perform a Interlocked Read to avoid an
            ;3845      ; Interlock Sequence Fault.
U 127C, 081B,001C,1980,0800,0000,127D  ;3846  D_NOT.K[ZERO]  ; Load D with xFFFFFFFF for lockwrite
U 127D, 0001,003C,0180,09E0,0000,127E  ;3847  RC[0C]_D  ; Save write data in RC[0C] in case error
U 127E, 0000,003C,0180,B800,0000,1140  ;3848  MCT/LOCKWRITE.P  ; Start the lock write
            ;3849 =0 CALL[WAIT.LOOP],
U 1140, 0810,0039,0180,0968,0010,1249  ;3850      D_RC[0D],CLK.UBCC ; Wait in single step mode incase timeout
            ;3851  ;
            ;3852  ; End of single time state mode
            ;3853  ;
U 1141, 0000,003C,0180,0800,0000,106C  ;3854  NOP
U 106C, 0000,003D,8580,0800,0084,724A  ;3855 =00 CHECK.SBI.ERR[.C] ; Was there a time out on the write ?
```

```
U 106D, 0000,003C,0180,0800,0000,1070  ;3856  J/T.24.S3.NOTMO  ; No timeout
```

```
            ;3857  .PAGE
            ;3858 ;
            ;3859 ;////////////////////////////////////////////////////////////////////////////
            ;3860 ;
            ;3861 ; Got time out ( SBI Error Bit <12> ) after write lock. Call an error.
            ;3862 ;
U 106E, 0000,003D,5D80,0800,0084,70BE  ;3863  ERROR2[.7]
U 106F, 0000,003C,0180,0800,0000,1070  ;3864  NOP
            ;3865 ;
            ;3866 ;////////////////////////////////////////////////////////////////////////////
            ;3867 ;
            ;3868 ; ERROR LOGOUT:
            ;3869 ;
            ;3870 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3871 ;  Modified Cycle Count
            ;3872 ;  NOT USED
            ;3873 ;  NOT USED
            ;3874 ;  Address Under Test
            ;3875 ;  Cycle Count for Single Stepping
            ;3876 ;  ID[sbi.err] Register
            ;3877 ;
            ;3878 ;////////////////////////////////////////////////////////////////////////////
            ;3879 ;
            ;3880  ;
            ;3881  ; Check for any unexpected interrupts.
            ;3882  ;
            ;3883 =00
            ;3884 T.24.S3.NOTMD:
U 1070, 0000,003D,0180,0800,0000,11F3  ;3885  CHECK.INTERRUPT  ; Check for interrupts
U 1071, 0000,003C,0180,0800,0000,1073  ;3886  J/T.24.S3.NOINT  ; No interrupt
```

```
                ;3887   .PAGE
                ;3888   ;
                ;3889   ;////////////////////////////////////////////////////////////////////////////////
                ;3890   ;
                ;3891   ; Got an unexpected interrupt after lockwrite.  Call an error.
                ;3892   ;
U 1072, 0000,003D,F580,0800,0084,70BE   ;3893   ERROR2[.A]
                ;3894   ;
                ;3895   ;////////////////////////////////////////////////////////////////////////////////
                ;3896   ;
                ;3897   ; ERROR LOGOUT:
                ;3898   ;
                ;3899   ; DATA: I/O Base Address of MS780-E Currently under test
                ;3900   ;  Modified Cycle Count
                ;3901   ;  NOT USED
                ;3902   ;  NOT USED
                ;3903   ;  Address Under Test
                ;3904   ;  Cycle Count for Single Stepping
                ;3905   ;  ID[vector] Register
                ;3906   ;  ID[sbi.err] Register
                ;3907   ;  ID[fault] Register
                ;3908   ;  Interrupt Occurred Flag  1 = Interrupt Occurred
                ;3909   ;       0 = No Interrupt Occurred
                ;3910   ;
                ;3911   ;////////////////////////////////////////////////////////////////////////////////
                ;3912   ;
                ;3913    ;
                ;3914    ; Check the data to insure that it is correct
                ;3915    ;
                ;3916 T.24.S3.NOINT:
U 1073, 0010,0038,0180,0950,0200,127F   ;3917   VA_RC[0A]   ; Load VA with test address
U 127F, 0000,003C,0180,C800,0000,1280   ;3918   D[LONG]_CACHE.P   ; Read data from LOCKWRITE
U 1280, 0001,003C,0180,09D8,0000,1281   ;3919   RC[0B]_D   ; Save in RC[0B]
U 1281, 0019,0014,0580,0800,0010,1282   ;3920   ALU_D+K[.1],CLK.UBCC ; Add 1 to read data - should get 0's
U 1282, 0000,013C,0180,0800,0000,1142   ;3921   Z?
```

```
            ;3922  .PAGE
            ;3923 ;
            ;3924 ;////////////////////////////////////////////////////////////////////////////
            ;3925 ;
            ;3926 ; Got the wrong data when reading the data back.  Call an error.
            ;3927 ;
U 1142, 0000,003D,D580,0800,0084,70BE  ;3928 =0 ERROR2[.6]
            ;3929 ;
            ;3930 ;////////////////////////////////////////////////////////////////////////////
            ;3931 ;
            ;3932 ; ERROR LOGOUT:
            ;3933 ;
            ;3934 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3935 ;  Modified cycle count
            ;3936 ;  Expected Memory Data
            ;3937 ;  Received Memory Data
            ;3938 ;  Address Under Test
            ;3939 ;  Cycle Count for Single Stepping
            ;3940 ;
            ;3941 ;////////////////////////////////////////////////////////////////////////////
            ;3942 ;
U 1143, 0000,003C,0180,0800,0000,1144  ;3943  NOP
            ;3944 ;
            ;3945 ;============================================================================
            ;3946 ;
            ;3947 ; Subtest 4 Interlock Read
            ;3948 ;
U 1144, 0000,003D,0180,0800,0000,11CB  ;3949 =0 SUBTEST   ; Update the subtest counter
U 1145, 0000,003C,0180,0800,0000,1146  ;3950  NOP
U 1146, 0000,003D,0180,0800,0000,11C3  ;3951 =0 ERLOOP   ; Set the error loop
U 1147, 0000,003C,0180,0800,0000,1148  ;3952  NOP
U 1148, 0000,003D,0180,0800,0000,10CC  ;3953 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 1149, 0010,0038,0180,0950,0200,1283  ;3954  VA_RC[0A]  ; Load VA with test address
U 1283, 081B,001C,1980,0800,0000,1284  ;3955  D_NOT.K[ZERO]  ; Load xFFFFFFFF in D to init the test location
U 1284, 0001,003C,0180,09E0,0000,1285  ;3956  RC[0C]_D  ; Save write data in RC[0C] in case error
U 1285, 0000,003C,0180,A800,0000,114A  ;3957  CACHE.P_D[LONG]  ; Init to one's
U 114A, 0000,003D,8980,0800,0084,7206  ;3958 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 114B, 0000,003C,0180,0800,0000,114C  ;3959  NOP
            ;3960 =0 D_RC[09],
U 114C, 0810,0039,0180,0948,0000,124E  ;3961       CALL[STEP]  ; Start single stepping
            ;3962  ;
            ;3963  ; Following sequence is performed in single time state mode
            ;3964  ;
U 114D, 0000,003C,0180,E800,0000,1286  ;3965  MCT/LOCKREAD.P  ; Perform a read first
U 1286, 0001,003C,0180,09D8,0000,114E  ;3966  RC[0B]_D  ; Save data from read in RC[0B]
            ;3967 =0 CALL[WAIT.LOOP],
U 114E, 0810,0039,0180,0948,0010,1249  ;3968       D_RC[09],CLK.UBCC ; Wait in single step mode incase timeout
            ;3969  ;
            ;3970  ; End of single time state mode
            ;3971  ;
U 114F, 0000,003C,0180,0800,0000,1074  ;3972  NOP
U 1074, 0000,003D,0180,0800,0000,1200  ;3973 =00 CHECK.UTRAP  ; Was there a time out miro trap ?
U 1075, 0000,003C,0180,0800,0000,1078  ;3974  J/T.24.S4.NOTMO  ; No, check to see if any unexpected interrupts
```

```
            ;3975  .PAGE
            ;3976 ;
            ;3977 ;////////////////////////////////////////////////////////////////////////////////
            ;3978 ;
            ;3979 ; Got a time out micro trap on the lockread.  Call an error.
            ;3980 ;
U 1076, 0000,003D,0180,0800,0084,70BE  ;3981  ERROR2[.8]
U 1077, 0000,003C,0180,0800,0000,1078  ;3982  NOP
            ;3983 ;
            ;3984 ;////////////////////////////////////////////////////////////////////////////////
            ;3985 ;
            ;3986 ; ERROR LOGOUT:
            ;3987 ;
            ;3988 ; DATA: I/O Base Address of MS780-E Currently under test
            ;3989 ;  Unused
            ;3990 ;  Unused
            ;3991 ;  Unused
            ;3992 ;  Address Under Test
            ;3993 ;  Cycle Count for Single Stepping
            ;3994 ;  Expected Micro Trap Flags
            ;3995 ;  Received Micro Trap Flags
            ;3996 ;
            ;3997 ;////////////////////////////////////////////////////////////////////////////////
            ;3998 ;
            ;3999  ;
            ;4000  ; Check for any unexpected interrupts.
            ;4001  ;
            ;4002 =00
            ;4003 T.24.S4.NOTMO:
U 1078, 0000,003D,0180,0800,0000,11F3  ;4004  CHECK.INTERRUPT  ; Check for interrupts
U 1079, 0000,003C,0180,0800,0000,107B  ;4005  J/T.24.S4.NOINT  ; No interrupt
```

```
            ;4006  .PAGE
            ;4007 ;
            ;4008 ;////////////////////////////////////////////////////////////////////////////////
            ;4009 ;
            ;4010 ; Got an unexpected interrupt on the lockread.  Call an error.
            ;4011 ;
U 107A, 0000,003D,F580,0800,0084,70BE  ;4012  ERROR2[.A]
            ;4013 ;
            ;4014 ;////////////////////////////////////////////////////////////////////////////////
            ;4015 ;
            ;4016 ; ERROR LOGOUT:
            ;4017 ;
            ;4018 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4019 ;  Unused
            ;4020 ;  Unused
            ;4021 ;  Unused
            ;4022 ;  Address Under Test
            ;4023 ;  Cycle Count for Single Stepping
            ;4024 ;  ID[vector] Register
            ;4025 ;  ID[sbi.err] Register
            ;4026 ;  ID[fault] Register
            ;4027 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4028 ;      0 = No Interrupt Occurred
            ;4029 ;
            ;4030 ;////////////////////////////////////////////////////////////////////////////////
            ;4031 ;
            ;4032  ;
            ;4033  ; Check the data to insure that it is correct.
            ;4034  ;
            ;4035 T.24.S4.NOINT:
U 107B, 0810,0038,0180,0958,0000,1287  ;4036  D_RC[0B]  ; Load the D with the data from the lockread
U 1287, 0019,0014,0580,0800,0010,1288  ;4037  ALU_D←K[.1],CLK.UBCC ; Was data written ok?
U 1288, 0000,013C,0180,0800,0090,1150  ;4038  Z?   ; Branch if yes
```

```
            ;4039  .PAGE
            ;4040 ;
            ;4041 ;////////////////////////////////////////////////////////////////////////////////
            ;4042 ;
            ;4043 ; Got the wrong data from the lockread.  Call an error.
            ;4044 ;
U 1150, 0000,003D,D580,0800,0084,70BE  ;4045 =0 ERROR2[.6]
            ;4046 ;
            ;4047 ;////////////////////////////////////////////////////////////////////////////////
            ;4048 ;
            ;4049 ; ERROR LOGOUT:
            ;4050 ;
            ;4051 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4052 ;  Unused
            ;4053 ;  Expected Memory Data
            ;4054 ;  Received Memory Data
            ;4055 ;  Address Under Test
            ;4056 ;  Cycle Count for Single Stepping
            ;4057 ;
            ;4058 ;////////////////////////////////////////////////////////////////////////////////
            ;4059 ;
            ;4060  ;
            ;4061  ; Go back to the Restart entry point and test another MS780-E
            ;4062  ;
U 1151, 0000,003C,0180,0800,0000,1050  ;4063  J/RESTART.TEST.24  ; Go see if more MS780-E's to test
            ;4064  ;
            ;4065  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!
            ;4066  ;
            ;4067 T.24.EXIT:
U 1289, 0000,003C,0180,0800,0000,1152  ;4068  NOP
            ;4069
```

```
;4070  .PAGE "TEST 1AF SINGLE BUS STATE SBI INTERFACE REFERENCES"
;4071 ;
;4072 ;*****************************************************************************
;4073 ;++
;4074 ; Functional Description:
;4075 ; ----------------------
;4076 ;
;4077 ; This test checks whether the SBI Interface can perfrom write/read
;4078 ; operations in Single BUS Cycle mode.
;4079 ;
;4080 ;   Subtest 1
;4081 ;
;4082 ; The Data Latches are written with a test pattern while the CPU is being
;4083 ; single stepped by the LSI-11.  A normal read is performed to check data
;4084 ; integrity.
;4085 ;
;4086 ;   Subtest 2
;4087 ;
;4088 ; A normal write operation is made to the Data Latches.  A Read will
;4089 ; follow with the CPU being single cycled, and the data retreived will
;4090 ; be check for errors.
;4091 ;
;4092 ; -----------------
;4093 ;
;4094 ;
;4095 ; Error Logout:
;4096 ; ------------
;4097 ;
;4098 ; See individual error reports.
;4099 ;
;4100 ; ----------------------
;4101 ;
;4102 ;
;4103 ;=============================================================================
;4104 ;
;4105 ; Register Map:
;4106 ; ------------
;4107 ;
;4108 ; RA,RB  Description
;4109 ; -----  -----------
;4110 ;
;4111 ;    Not Used
;4112 ;
;4113 ; RC  Description
;4114 ; --  -----------
;4115 ;
;4116 ; E  I/O Base Address of MS780-E Currently Under Test
;4117 ;
;4118 ; C  Expected Memory Data
;4119 ;
;4120 ; B  Received Memory Data
;4121 ;
;4122 ; A  Address  Under Test
;4123 ;
;4124 ; 9  Cycle Count for Stepping CPU
```

```
            ;4125 ;
            ;4126 ;--
            ;4127 =0
U 1152, 0000,003D,0180,0800,0000,11AE  ;4128 T.25: NEWTST   ; Signify start of new test
U 1153, 0000,003C,0180,0800,0000,107C  ;4129  NOP
U 107C, 0000,003D,0180,0800,0000,10D0  ;4130 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
            ;4131     ; (Implicit enable of SBI)
U 107D, 0000,003C,0180,0800,0000,1093  ;4132  J/T.25.EXIT  ; No memories
            ;4133  ;
            ;4134  ; This is the point (RESTART.TEST.25) where after testing the functions
            ;4135  ; the program will return to here if there is another MS780-E on the
            ;4136  ; system and repeat the sequence of instructions.
            ;4137  ;
            ;4138 RESTART.TEST.25:
U 107E, 0000,003C,D980,0800,0084,707F  ;4139  SC_K[.9]  ; SC gets x9 for mask of bit 9
U 107F, 0803,001C,0180,0800,0000,128A  ;4140  D_NOT.MASK  ; Load the value x200 into the D register
U 128A, 0001,003C,0180,09C8,0000,128B  ;4141  RC[09]_D  ; Save x200 in RC[09]
U 128B, 0010,0038,01C0,0970,0000,128C  ;4142  Q_RC[0E]  ; Move MS780-E base address to Q
U 128C, 0019,2014,6580,09D0,0000,1154  ;4143  RC[0A]_Q+K[.10]  ; Point address to Data Latch 1
            ;4144 ;
            ;4145 ;================================================================================
            ;4146 ;
            ;4147 ; Subtest 1 Register Writes
            ;4148 ;
U 1154, 0000,003D,0180,0800,0000,11CB  ;4149 =0 SUBTEST   ; Increment Subtest count
U 1155, 0000,003C,0180,0800,0000,1156  ;4150  NOP
U 1156, 0000,003D,0180,0800,0000,11C3  ;4151 =0 ERLOOP   ; Loop on error from here
U 1157, 0000,003C,0180,0800,0000,1158  ;4152  NOP
U 1158, 0000,003D,0180,0800,0000,10CC  ;4153 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1159, 0010,0038,0180,0950,0200,128D  ;4154  VA_RC[0A]  ; Load VA with address of Data latch 1
U 128D, 0F00,003C,0180,0800,0000,128E  ;4155  D_0  ; Init contents of data latch 1
U 128E, 0000,003C,0180,A800,0000,128F  ;4156  CACHE.P_D[LONG]  ; to zero
U 128F, 0810,0038,0180,0948,0000,115A  ;4157  D_RC[09]  ; Load D with the number of cycles to step
U 115A, 0000,003D,0180,0800,0000,1251  ;4158 =0 CALL[BUS]  ; Start single cycling
            ;4159  ;
            ;4160  ; Following sequence is performed in single cycle mode
            ;4161  ;
U 115B, 081B,001C,1980,0800,0000,1290  ;4162  D_NOT.K[ZERO]  ; Setup to write all ones to data latch 1
U 1290, 0001,003C,0180,09E0,0000,1291  ;4163  RC[0C]_D  ; Save write data in RC[0C] in case error
U 1291, 0000,003C,0180,A800,0000,115C  ;4164  CACHE.P_D[LONG]  ; Start the write
            ;4165 =0 CALL[WAIT.LOOP],
U 115C, 0810,0039,0180,0948,0010,1249  ;4166      D_RC[09],CLK.UBCC ; Wait in single cycle mode incase timeout
            ;4167  ;
            ;4168  ; End of single cycle mode
            ;4169  ;
U 115D, 0000,003C,0180,0800,0000,1080  ;4170  NOP
U 1080, 0000,003D,8580,0800,0084,724A  ;4171 =00 CHECK.SBI.ERR[.C] ; Was there a timeout on the write ?
U 1081, 0000,003C,0180,0800,0000,1084  ;4172  J/T.25.S1.NOTMO  ; No, go check for any unexpected interrupts
```

```
            ;4173  .PAGE
            ;4174 ;
            ;4175 ;////////////////////////////////////////////////////////////////////////////
            ;4176 ;
            ;4177 ; Got a time out ( SBI Error Bit <12> ) on the write.  Call an error
            ;4178 ;
U 1082, 0000,003D,5D80,0800,0084,70BE  ;4179 =0 ERROR2[.7]
U 1083, 0000,003C,0180,0800,0000,1084  ;4180  NOP
            ;4181 ;
            ;4182 ;////////////////////////////////////////////////////////////////////////////
            ;4183 ;
            ;4184 ; ERROR LOGOUT:
            ;4185 ;
            ;4186 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4187 ;  CNFG C/D Address of MS780-E Currently under test
            ;4188 ;  NOT USED
            ;4189 ;  NOT USED
            ;4190 ;  Address Under Test
            ;4191 ;  Cycle Count for Single Stepping
            ;4192 ;  ID[sbi.err] Register
            ;4193 ;
            ;4194 ;////////////////////////////////////////////////////////////////////////////
            ;4195 ;
            ;4196  ;
            ;4197  ; Check for unexpected Interrupt.
            ;4198  ;
            ;4199 =00
            ;4200 T.25.S1.NOTMO:
U 1084, 0000,003D,0180,0800,0000,11F3  ;4201  CHECK.INTERRUPT  ; Any unexpected interrupts ?
U 1085, 0000,003C,0180,0800,0000,1087  ;4202  J/T.25.S1.NOINT  ; No interrupt
```

```
            ;4203  .PAGE
            ;4204 ;
            ;4205 ;////////////////////////////////////////////////////////////////////////////////
            ;4206 ;
            ;4207 ; Got an unexpected interrupt on the write.  Call an error
            ;4208 ;
U 1086, 0000.003D.F580.0800.0084.70BE  ;4209  ERROR2[.A]
            ;4210 ;
            ;4211 ;////////////////////////////////////////////////////////////////////////////////
            ;4212 ;
            ;4213 ; ERROR LOGOUT:
            ;4214 ;
            ;4215 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4216 ;  CNFG C/D Address of MS780-E Currently under test
            ;4217 ;  NOT USED
            ;4218 ;  NOT USED
            ;4219 ;  Address Under Test
            ;4220 ;  Cycle Count for Single Stepping
            ;4221 ;  ID[vector] Register
            ;4222 ;  ID[sbi.err] Register
            ;4223 ;  ID[fault] Register
            ;4224 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4225 ;      0 = No Interrupt Occurred
            ;4226 ;
            ;4227 ;////////////////////////////////////////////////////////////////////////////////
            ;4228 ;
            ;4229  ;
            ;4230  ; Check the Data to see if it is correct
            ;4231  ;
            ;4232 T.25.S1.NOINT:
U 1087, 0010.0038.0180.0950.0200.1292  ;4233  VA_RC[0A]  ; Load VA with test address
U 1292, 0000.003C.0180.C800.0000.1293  ;4234  D[LONG]_CACHE.P  ; Read data latch 1
U 1293, 0001.003C.0180.09D8.0000.1294  ;4235  RC[0B]_D  ; Store data read in case of an error
U 1294, 0019.0014.0580.0800.0010.1295  ;4236  ALU_D+K[.1],CLK.UBCC ; Was latch written ok?
U 1295, 0000.013C.0180.0800.0000.115E  ;4237  Z?  ; Branch if yes
```

```
            ;4238  .PAGE
            ;4239 ;
            ;4240 ;////////////////////////////////////////////////////////////////////////////////
            ;4241 ;
            ;4242 ; Got the wrong data when reading the data back.  Call an error
            ;4243 ;
U 115E, 0000.003D.D580.0800.0084.70BE  ;4244 =0 ERROR2[.6]
U 115F, 0000.003C.0180.0800.0000.1160  ;4245  NOP
            ;4246 ;
            ;4247 ;////////////////////////////////////////////////////////////////////////////////
            ;4248 ;
            ;4249 ; ERROR LOGOUT:
            ;4250 ;
            ;4251 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4252 ;  CNFG C/D Address of MS780-E Currently under test
            ;4253 ;  Expected Memory Data
            ;4254 ;  Received Memory Data
            ;4255 ;  Address Under Test
            ;4256 ;  Cycle Count for Single Stepping
            ;4257 ;
            ;4258 ;////////////////////////////////////////////////////////////////////////////////
            ;4259 ;
            ;4260 ;
            ;4261 ;================================================================================
            ;4262 ;
            ;4263 ; Subtest 2 Register Read
            ;4264 ;
U 1160, 0000.003D.0180.0800.0000.11CB  ;4265 =0 SUBTEST   ; Increment subtest counter
U 1161, 0000.003C.0180.0800.0000.1162  ;4266  NOP
U 1162, 0000.003D.0180.0800.0000.11C3  ;4267 =0 ERLOOP   ; Set the error loop
U 1163, 0000.003C.0180.0800.0000.1164  ;4268  NOP
U 1164, 0000.003D.0180.0800.0000.10CC  ;4269 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 1165, 0010.0038.0180.0950.0200.1296  ;4270  VA_RC[0A]  ; Load VA with the test address
U 1296, 081B.001C.1980.0800.0000.1297  ;4271  D_NOT.K[ZERO]
U 1297, 0000.003C.0180.A800.0000.1166  ;4272  CACHE.P_D[LONG]  ; to zero
U 1166, 0000.003D.8980.0800.0084.7206  ;4273 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 1167, 0000.003C.0180.0800.0000.1298  ;4274  NOP
U 1298, 0810.0038.0180.0948.0000.1168  ;4275  D_RC[09]  ; D gets the number of cycles to single step
U 1168, 0000.003D.0180.0800.0000.1251  ;4276 =0 CALL[BUS]  ; Start single cycling
            ;4277  ;
            ;4278  ; Following sequence is performed in single cycle mode
            ;4279  ;
U 1169, 0000.003C.0180.C800.0000.1299  ;4280  D[LONG]_CACHE.P  ; Start the read
U 1299, 0001.003C.0180.09D8.0000.116A  ;4281  RC[0B]_D  ; Save read data
            ;4282 =0 CALL[WAIT.LOOP],
U 116A, 0810.0039.0180.0948.0010.1249  ;4283      D_RC[09],CLK.UBCC ; Wait in single cycle mode incase timeout
            ;4284  ;
            ;4285  ; End of single cycle mode
            ;4286  ;
U 116B, 0000.003C.0180.0800.0000.1088  ;4287  NOP
U 1088, 0000.003D.0180.0800.0000.1200  ;4288 =00 CHECK.UTRAP  ; Was there a Time Out micro trap ?
U 1089, 0000.003C.0180.0800.0000.108C  ;4289  J/T.25.S2.NOTMO  ; No, check for unexpected interrupts
```

```
            ;4290  .PAGE
            ;4291 ;
            ;4292 ;////////////////////////////////////////////////////////////////////////////////
            ;4293 ;
            ;4294 ; Got a time out micro trap on the read from the register.  Call an error.
            ;4295 ;
U 108A, 0000.003D.0180.0800.0084.70BE  ;4296  ERROR2[.8]
U 108B, 0000.003C.0180.0800.0000.108C  ;4297  NOP
            ;4298 ;
            ;4299 ;////////////////////////////////////////////////////////////////////////////////
            ;4300 ;
            ;4301 ; ERROR LOGOUT:
            ;4302 ;
            ;4303 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4304 ;  CNFG C/D Address of MS780-E Currently under test
            ;4305 ;  NOT USED
            ;4306 ;  NOT USED
            ;4307 ;  Address Under Test
            ;4308 ;  Cycle Count for Single Stepping
            ;4309 ;  Expected Micro Trap Flags
            ;4310 ;  Received Micro Trap Flags
            ;4311 ;
            ;4312 ;////////////////////////////////////////////////////////////////////////////////
            ;4313 ;
            ;4314  ;
            ;4315  ; Check for any unexpected interrupts
            ;4316  ;
            ;4317 =00
            ;4318 T.25.S2.NOTMO:
U 108C, 0000.003D.0180.0800.0000.11F3  ;4319  CHECK.INTERRUPT  ; Check for interrupts
U 108D, 0000.003C.0180.0800.0000.108F  ;4320  J/T.25.S2.NOINT  ; No interrupt
```

```
            ;4321   .PAGE
            ;4322  ;
            ;4323  ;////////////////////////////////////////////////////////////////////////////
            ;4324  ;
            ;4325  ; Got an unexpected interrupt on the read.  Call an error.
            ;4326  ;
U 108E, 0000,003D,F580,0800,0084,70BE   ;4327  ERROR2[.A]
            ;4328  ;
            ;4329  ;////////////////////////////////////////////////////////////////////////////
            ;4330  ;
            ;4331  ; ERROR LOGOUT:
            ;4332  ;
            ;4333  ; DATA: I/O Base Address of MS780-E Currently under test
            ;4334  ;  CNFG C/D Address of MS780-E Currently under test
            ;4335  ;  NOT USED
            ;4336  ;  NOT USED
            ;4337  ;  Address Under Test
            ;4338  ;  Cycle Count for Single Stepping
            ;4339  ;  ID[vector] Register
            ;4340  ;  ID[sbi.err] Register
            ;4341  ;  ID[fault] Register
            ;4342  ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4343  ;       0 = No Interrupt Occurred
            ;4344  ;
            ;4345  ;////////////////////////////////////////////////////////////////////////////
            ;4346  ;
            ;4347   ;
            ;4348   ; check the data to insure that it is correct.
            ;4349   ;
            ;4350 T.25.S2.NOINT:
U 108F, 0810,0038,0180,0958,0000,129A   ;4351  D_RC[0B]   ; Move read data to D for validity check
U 129A, 0019,0014,0580,0800,0010,129B   ;4352  ALU_D+K[.1],CLK.UBCC ; Was latch written ok?
U 129B, 0000,013C,0180,0800,0000,116C   ;4353  Z?    ; Branch if data ok
```

```
            ;4354  .PAGE
            ;4355 ;
            ;4356 ;////////////////////////////////////////////////////////////////////////////////
            ;4357 ;
            ;4358 ; Data read was not correct.  Call an error.
            ;4359 ;
U 116C, 0000.003D.D580.0800.0084.70BE  ;4360 =0 ERROR2[.6]
U 116D, 0000.003C.0180.0800.0000.1090  ;4361  NOP
            ;4362 ;
            ;4363 ;////////////////////////////////////////////////////////////////////////////////
            ;4364 ;
            ;4365 ; ERROR LOGOUT:
            ;4366 ;
            ;4367 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4368 ;  Unused
            ;4369 ;  Expected Memory Data
            ;4370 ;  Received Memory Data
            ;4371 ;  Address Under Test
            ;4372 ;  Cycle Count for Single Stepping
            ;4373 ;
            ;4374 ;////////////////////////////////////////////////////////////////////////////////
            ;4375 ;
            ;4376  ;
            ;4377  ; At this point we have complete the bus functions under test.  The
            ;4378  ; next thing is to try enabling the next MS780-E and if there is indeed
            ;4379  ; another one then go to RESTART.TEST.25 and run test over, if not then
            ;4380  ; exit this test.
            ;4381  ;
U 1090, 0000.003D.0180.0800.0000.123B  ;4382 =00 CALL[ENABLE.NEXT.MS780E] ; Check and see if more MS780-E's ?
U 1091, 0000.003C.0180.0800.0000.107E  ;4383  J/RESTART.TEST.25  ; Yes, more to test, restart test
U 1092, 0000.003C.0180.0800.0000.1093  ;4384  NOP    ; No more to test, exit
            ;4385  ;
            ;4386  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!
            ;4387  ;
            ;4388 T.25.EXIT:
U 1093, 0000.003C.0180.0800.0000.116E  ;4389  NOP
            ;4390
```

```
;4391   .PAGE "TEST 1B0 SINGLE BUS STATE ARRAY/SBI INTERFACE REF."
;4392 ;
;4393 ;*****************************************************************************
;4394 ;++
;4395 ; Functional Description:
;4396 ; ----------------------
;4397 ;
;4398 ; This test makes use of a routine executing in the monitor (on the
;4399 ; LSI-11) to single cycle the CPU.  Its main function is to verify that
;4400 ; array references are executed properly when the CPU is in the single
;4401 ; BUS mode.  No interrupts or uTraps are expected.  The following
;4402 ; functions are checked:
;4403 ;
;4404 ;    i.  Memory Write
;4405 ;   ii.  Memory Read
;4406 ;  iii.  Interlocked Write
;4407 ;   iv.  Interlocked Read
;4408 ;
;4409 ;   Subtest 1
;4410 ;
;4411 ; This subtest writes some data pattern to the array in Single CPU State.
;4412 ; The write is followed by a "normal" read operation to check the
;4413 ; integrity of the data.
;4414 ;
;4415 ;   Subtest 2
;4416 ;
;4417 ; This subtest initializes the array to some known data pattern and then
;4418 ; executes a read cycle in single Bus state.  The data returned is
;4419 ; checked for integrity against the original pattern.
;4420 ;
;4421 ;   Subtest 3
;4422 ;
;4423 ; This subtest is similar to Subtest 1, except that the memory cycle will
;4424 ; be an interlock write.  The array will be initialized with an Interlock
;4425 ; Read to avoid generating a Interlock Sequence Fault.
;4426 ;
;4427 ;   Subtest 4
;4428 ;
;4429 ; This subtest is similar to Subtest 2, except that the memory cycle
;4430 ; executed in Single Bus State will be an Interlock Read.
;4431 ;
;4432 ; -----------------
;4433 ;
;4434 ;
;4435 ; Error Logout:
;4436 ; ------------
;4437 ;
;4438 ; See individual error reports.
;4439 ;
;4440 ; ----------------------
;4441 ;
;4442 ;
;4443 ;=============================================================================
;4444 ;
;4445 ; Register Map:
```

```
            ;4446 ; ------------
            ;4447 ;
            ;4448 ; RA,RB  Description
            ;4449 ; -----  -----------
            ;4450 ;
            ;4451 ;      Not Used
            ;4452 ;
            ;4453 ; RC  Description
            ;4454 ; --  -----------
            ;4455 ;
            ;4456 ; E  I/O Base Address of MS780-E Currently Under Test
            ;4457 ;
            ;4458 ; C  Expected Memory Data
            ;4459 ;
            ;4460 ; B  Received Memory Data
            ;4461 ;
            ;4462 ; A  Address  Under Test
            ;4463 ;
            ;4464 ; 9  Cycle Count for Stepping CPU
            ;4465 ;
            ;4466 ;--
            ;4467 ;*****************************************************************************
            ;4468 =0
U 116E, 0000.003D.0180.0800.0000.11AE  ;4469 T.26: NEWTST
U 116F, 0000.003C.0180.0800.0000.1094  ;4470  NOP
U 1094, 0000.003D.0180.0800.0000.10D0  ;4471 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1095, 0000.003C.0180.0800.0000.12BE  ;4472  J/T.26.EXIT  ; No memories, exit this test
U 1096, 0000.003C.0180.0800.0000.1098  ;4473  NOP
            ;4474 =
            ;4475  ;
            ;4476  ; This is the point where the program loops back to after executing one
            ;4477  ; pass of this test, if there are more MS780-E's to test on the system.
            ;4478  ;
            ;4479 =00
            ;4480 RESTART.TEST.26:
U 1098, 0000.003D.0180.0800.0000.10C2  ;4481  CALL[START.TEST] ; Configure the MS780-E
U 1099, 0000.003C.0180.0800.0000.12BE  ;4482  J/T.26.EXIT  ; Couldn't configure or no more left
U 109A, 0000.003C.D980.0800.0084.709B  ;4483  SC_K[.9]  ; SC gets x9 for mask of bit 9
U 109B, 0803.001C.0180.0800.0000.129C  ;4484  D_NOT.MASK  ; Load the value x200 into the D register
U 129C, 0001.003C.0180.09C8.0000.129D  ;4485  RC[09]_D  ; Save in RC[09]
U 129D, 0018.0038.1980.09D0.0000.1170  ;4486  RC[0A]_K[ZERO]  ; Set-up RC[0A] with address 0 for test
            ;4487 ;
            ;4488 ;=============================================================================
            ;4489 ;
            ;4490 ; Subtest 1 Array Writes
            ;4491 ;
U 1170, 0000.003D.0180.0800.0000.11CB  ;4492 =0 SUBTEST   ; Set subtest counter
U 1171, 0000.003C.0180.0800.0000.1172  ;4493  NOP
U 1172, 0000.003D.0180.0800.0000.11C3  ;4494 =0 ERLOOP   ; Loop on error from here
U 1173, 0000.003C.0180.0800.0000.1174  ;4495  NOP
U 1174, 0000.003D.0180.0800.0000.10CC  ;4496 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1175, 0010.0038.0180.0950.0200.129E  ;4497  VA_RC[0A]  ; Point VA to correct address
U 129E, 0F00.003C.0180.0800.0000.129F  ;4498  D_0  ; Init the D reg to 0 to init memory location
U 129F, 0000.003C.0180.A800.0000.1176  ;4499  CACHE.P_D[LONG]  ; Initialize the memory location to zero
            ;4500 =0 D_RC[09],
```

```
U 1176, 0810.0039.0190.0948.0000.1251  ;4501      CALL[BUS]  ; Start single cycling
        ;4502  ;
        ;4503  ; Following sequence is performed in single cycle mode
        ;4504  ;
U 1177, 081B.001C.1980.0800.0000.12A0  ;4505  D_NOT.K[ZERO]  ; Setup to write all ones to array
U 12A0, 0001.003C.0180.09E0.0000.12A1  ;4506  RC[0C]_D  ; Save write data in RC[0C] in case error
U 12A1, 0000.003C.0180.A800.0000.1178  ;4507  CACHE.P_D[LONG]  ; Start the write
        ;4508 =0 CALL[WAIT.LOOP],
U 1178, 0810.0039.0180.0948.0010.1249  ;4509      D_RC[09],CLK.UBCC ; Wait in single cycle mode incase timeout
        ;4510  ;
        ;4511  ; End of single cycle mode
        ;4512  ;
U 1179, 0000.003C.0180.0800.0000.109C  ;4513  NOP
U 109C, 0000.003D.8580.0800.0084.724A  ;4514 =00 CHECK.SBI.ERR[.C] ; Was there a time out on the write ?
U 109D, 0000.003C.0180.0800.0000.10A0  ;4515  J/T.26.S1.NOTMO  ; No timeout
```

```
            ;4516  .PAGE
            ;4517 ;
            ;4518 ;////////////////////////////////////////////////////////////////////////////////
            ;4519 ;
            ;4520 ; Got a time out ( SBI Error Bit <12> ) on the write to the memory location.
            ;4521 ; Call an error.
            ;4522 ;
U 109E, 0000,003D,5D80,0800,0084,70BE  ;4523  ERROR2[.7]
U 109F, 0000,003C,0180,0800,0000,10A0  ;4524  NOP
            ;4525 ;
            ;4526 ;////////////////////////////////////////////////////////////////////////////////
            ;4527 ;
            ;4528 ; ERROR LOGOUT:
            ;4529 ;
            ;4530 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4531 ;  CNFG C/D Address of MS780-E Currently under test
            ;4532 ;  NOT USED
            ;4533 ;  NOT USED
            ;4534 ;  Address Under Test
            ;4535 ;  Cycle Count for Single Stepping
            ;4536 ;  ID[sbi.err] Register
            ;4537 ;
            ;4538 ;////////////////////////////////////////////////////////////////////////////////
            ;4539 ;
            ;4540  ;
            ;4541  ; Check for unexpected interrupts.
            ;4542  ;
            ;4543 =00
            ;4544 T.26.S1.NOTMO:
U 10A0, 0000,003D,0180,0800,0000,11F3  ;4545  CHECK.INTERRUPT  ; Check for interrupts
U 10A1, 0000,003C,0180,0800,0000,10A3  ;4546  J/T.26.S1.NOINT  ; No interrupt
```

```
            ;4547   .PAGE
            ;4548  ;
            ;4549  ;////////////////////////////////////////////////////////////////////////////////
            ;4550  ;
            ;4551  ; Got an unexpected interrupt on the write.  Call an error.
            ;4552  ;
U 10A2, 0000,003D,F580,0800,0084,70BE   ;4553   ERROR2[.A]
            ;4554  ;
            ;4555  ;////////////////////////////////////////////////////////////////////////////////
            ;4556  ;
            ;4557  ; ERROR LOGOUT:
            ;4558  ;
            ;4559  ; DATA: I/O Base Address of MS780-E Currently under test
            ;4560  ;  CNFG C/D Address of MS780-E Currently under test
            ;4561  ;  NOT USED
            ;4562  ;  NOT USED
            ;4563  ;  Address Under Test
            ;4564  ;  Cycle Count for Single Stepping
            ;4565  ;  ID[vector] Register
            ;4566  ;  ID[sbi.err] Register
            ;4567  ;  ID[fault] Register
            ;4568  ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4569  ;       0 = No Interrupt Occurred
            ;4570  ;
            ;4571  ;////////////////////////////////////////////////////////////////////////////////
            ;4572  ;
            ;4573   ;
            ;4574   ; Check the Data to insure that it is correct.
            ;4575   ;
            ;4576  T.26.S1.NOINT:
U 10A3, 0010,0038,0180,0950,02C0,12A2   ;4577   VA_RC[0A]  ; Load VA with the test address
U 12A2, 0000,003C,0180,C800,0000,12A3   ;4578   D[LONG]_CACHE.P  ; Read the array
U 12A3, 0001,003C,0180,09D8,0000,12A4   ;4579   RC[0B]_D  ; Store the data read in case of an error
U 12A4, 0019,0014,0580,0800,0010,12A5   ;4580   ALU_D+K[.1],CLK.UBCC ; Was data written ok?
U 12A5, 0000,013C,0180,0800,0000,117A   ;4581   Z?   ; Branch if yes
```

```
            ;4582  .PAGE
            ;4583 ;
            ;4584 ;////////////////////////////////////////////////////////////////////////////
            ;4585 ;
            ;4586 ; Got the wrong data when reading the data back.  Call an error.
            ;4587 ;
U 117A, 0000,003D,D580,0800,0084,70BE  ;4588 =0 ERROR2[.6]
U 117B, 0000,003C,0180,0800,0000,117C  ;4589  NOP
            ;4590 ;
            ;4591 ;////////////////////////////////////////////////////////////////////////////
            ;4592 ;
            ;4593 ; ERROR LOGOUT:
            ;4594 ;
            ;4595 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4596 ;  Unused
            ;4597 ;  Expected Memory Data
            ;4598 ;  Received Memory Data
            ;4599 ;  Address Under Test
            ;4600 ;  Cycle Count for Single Stepping
            ;4601 ;
            ;4602 ;////////////////////////////////////////////////////////////////////////////
            ;4603 ;
            ;4604
            ;4605 ;
            ;4606 ;============================================================================
            ;4607 ;
            ;4608 ; Subtest 2 Array Read
            ;4609 ;
U 117C, 0000,003D,0180,0800,0000,11CB  ;4610 =0 SUBTEST   ; Increment subtest counter
U 117D, 0000,003C,0180,0800,0000,117E  ;4611  NOP
U 117E, 0000,003D,0180,0800,0000,11C3  ;4612 =0 ERLOOP   ; Set the error loop
U 117F, 0000,003C,0180,0800,0000,1180  ;4613  NOP
U 1180, 0000,003D,0180,0800,0000,10CC  ;4614 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 1181, 0010,0038,0180,0950,0200,12A6  ;4615  VA_RC[0A]  ; Load VA with test address
U 12A6, 081B,001C,1980,0800,0000,12A7  ;4616  D_NOT.K[ZERO]  ; Load D register with xFFFFFFFF and init
U 12A7, 0001,003C,0180,09E0,0000,12A8  ;4617  RC[0C]_D  ; Save write data in RC[0C] in case error
U 12A8, 0000,003C,0180,A800,0000,1182  ;4618  CACHE.P_D[LONG]  ; the memory location
U 1182, 0000,003D,8980,0800,0084,7206  ;4619 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 1183, 0000,003C,0180,0800,0000,1184  ;4620  NOP
            ;4621 =0 D_RC[09],
U 1184, 0810,0039,0180,0948,0000,1251  ;4622      CALL[BUS]  ; Start single cycling
            ;4623  ;
            ;4624  ; Following sequence is performed in single cycle mode
            ;4625  ;
U 1185, 0000,003C,0180,C800,0000,12A9  ;4626  D[LONG]_CACHE.P  ; Start the read
U 12A9, 0001,003C,0180,09D8,0000,1186  ;4627  RC[0B]_D  ; Save read data
            ;4628 =0 CALL[WAIT.LOOP],
U 1186, 0810,0039,0180,0948,0010,1249  ;4629      D_RC[09],CLK.UBCC ; Wait in single cycle mode incase timeout
            ;4630  ;
            ;4631  ; End of single cycle mode
            ;4632  ;
U 1187, 0000,003C,0180,0800,0000,10A4  ;4633  NOP
U 10A4, 0000,003D,0180,0800,0000,1200  ;4634 =00 CHECK.UTRAP  ; Did we get a time out micro trap ?
U 10A5, 0000,003C,0180,0800,0000,10A8  ;4635  J/T.26.S2.NOTMO  ; No, go check for any unexpected interrupts
```

```
            ;4636  .PAGE
            ;4637 ;
            ;4638 ;////////////////////////////////////////////////////////////////////////////////
            ;4639 ;
            ;4640 ; Got a time out micro trap on the read.  Call an error.
            ;4641 ;
U 10A6. 0000.003D.0180.0800.0084.70BE  ;4642  ERROR2[.8]
U 10A7. 0000.003C.0180.0800.0000.10A8  ;4643  NOP
            ;4644 ;
            ;4645 ;////////////////////////////////////////////////////////////////////////////////
            ;4646 ;
            ;4647 ; ERROR LOGOUT:
            ;4648 ;
            ;4649 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4650 ;  CNFG C/D Address of MS780-E Currently under test
            ;4651 ;  NOT USED
            ;4652 ;  NOT USED
            ;4653 ;  Address Under Test
            ;4654 ;  Cycle Count for Single Stepping
            ;4655 ;  Expected Micro Trap Flags
            ;4656 ;  Received Micro Trap Flags
            ;4657 ;
            ;4658 ;////////////////////////////////////////////////////////////////////////////////
            ;4659 ;
            ;4660  ;
            ;4661  ; Check for any unexpected interrupts.
            ;4662  ;
            ;4663 =00
            ;4664 T.26.S2.NOTMO:
U 10AB. 0000.003D.0180.0800.0000.11F3  ;4665  CHECK.INTERRUPT  ; Check for interrupts
U 10A9. 0000.003C.0180.0800.0000.10AB  ;4666  J/T.26.S2.NOINT  ; No interrupt
```

```
            ;4667  .PAGE
            ;4668 ;
            ;4669 ;////////////////////////////////////////////////////////////////////////////////
            ;4670 ;
            ;4671 ; Got an unexpected interrupt on the read.  Call an error.
            ;4672 ;
U 10AA, 0000,003D,F580,0800,0084,70BE  ;4673  ERROR2[.A]
            ;4674 ;
            ;4675 ;////////////////////////////////////////////////////////////////////////////////
            ;4676 ;
            ;4677 ; ERROR LOGOUT:
            ;4678 ;
            ;4679 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4680 ;  CNFG C/D Address of MS780-E Currently under test
            ;4681 ;  NOT USED
            ;4682 ;  NOT USED
            ;4683 ;  Address Under Test
            ;4684 ;  Cycle Count for Single Stepping
            ;4685 ;  ID[vector] Register
            ;4686 ;  ID[sbi.err] Register
            ;4687 ;  ID[fault] Register
            ;4688 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4689 ;       0 = No Interrupt Occurred
            ;4690 ;
            ;4691 ;////////////////////////////////////////////////////////////////////////////////
            ;4692 ;
            ;4693  ;
            ;4694  ; Check the data to insure that it is correct.
            ;4695  ;
            ;4696 T.26.S2.NOINT:
U 10AB, 0810,0038,0180,0958,0000,12AB  ;4697  D_RC[0B]  ; Move data read to D register
U 12AB, 0019,0014,0580,0800,0010,12AC  ;4698  ALU_D+K[.1],CLK.UBCC ; Add 1 to result and should get 0
U 12AC, 0000,013C,0180,0800,0000,1188  ;4699  Z?   ; Branch if data ok
```

```
            ;4700  .PAGE
            ;4701 ;
            ;4702 ;////////////////////////////////////////////////////////////////////////////////
            ;4703 ;
            ;4704 ; Got the wrong data when reading the data back.  Call an error.
            ;4705 ;
U 1188, 0000,003D,D580,0800,0084,70BE  ;4706 =0 ERROR2[.6]
            ;4707 ;
            ;4708 ;////////////////////////////////////////////////////////////////////////////////
            ;4709 ;
            ;4710 ; ERROR LOGOUT:
            ;4711 ;
            ;4712 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4713 ;  Unused
            ;4714 ;  Expected Memory Data
            ;4715 ;  Received Memory Data
            ;4716 ;  Address Under Test
            ;4717 ;  Cycle Count for Single Stepping
            ;4718 ;
            ;4719 ;////////////////////////////////////////////////////////////////////////////////
            ;4720 ;
U 1189, 0000,003C,0180,0800,0000,118A  ;4721  NOP
            ;4722 ;
            ;4723 ;================================================================================
            ;4724 ;
            ;4725 ; Subtest 3 Interlock Write
            ;4726 ;
U 118A, 0000,003D,0180,0800,0000,11CB  ;4727 =0 SUBTEST   ; Update Subtest counter
U 118B, 0000,003C,0180,0800,0000,118C  ;4728  NOP
U 118C, 0000,003D,0180,0800,0000,11C3  ;4729 =0 ERLOOP   ; Set the error loop
U 118D, 0000,003C,0180,0800,0000,118E  ;4730  NOP
U 118E, 0000,003D,0180,0800,0000,10CC  ;4731 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 118F, 0F10,0038,0180,0950,0200,12AD  ;4732  VA_RC[0A],D_0  ; Init contents of array
U 12AD, 0000,003C,0180,A800,0000,12AE  ;4733  CACHE.P_D[LONG]  ; to zero
U 12AE, 0810,0038,0180,0948,0000,12AF  ;4734  D_RC[09]  ; Move Cycle count to the D register
U 12AF, 0819,0014,8D80,0800,0000,12B0  ;4735  D_D+K[.1F]  ; Increase count to x21F to compensate for
            ;4736      ; doing the Lockread & lockwrite during step
U 12B0, 0001,003C,0180,09E8,0000,1190  ;4737  RC[0D]_D  ; Save modified cycle count in RC[0D]
U 1190, 0000,003D,0180,0800,0000,124E  ;4738 =0 CALL[STEP]  ; Start single stepping
            ;4739  ;
            ;4740  ; Following sequence is performed in single cycle mode
            ;4741  ;
U 1191, 0000,003C,0180,E800,0000,12B1  ;4742  MCT/LOCKREAD.P  ; Perform a Interlocked Read to avoid an
            ;4743      ; Interlock Sequence Fault.
U 12B1, 081B,001C,1980,0800,0000,12B2  ;4744  D_NOT.K[ZERO]  ; Load D with xFFFFFFFF for lockwrite
U 12B2, 0001,003C,0180,09E0,0000,12B3  ;4745  RC[0C]_D  ; Save write data in RC[0C] in case error
U 12B3, 0000,003C,0180,B800,0000,1192  ;4746  MCT/LOCKWRITE.P  ; Start the lock write
            ;4747 =0 CALL[WAIT.LOOP],
U 1192, 0810,0039,0180,0968,0010,1249  ;4748      D_RC[0D],CLK.UBCC ; Wait in single cycle mode incase timeout
            ;4749  ;
            ;4750  ; End of single cycle mode
            ;4751  ;
U 1193, 0000,003C,0180,0800,0000,10AC  ;4752  NOP
U 10AC, 0000,003D,8580,0800,0084,724A  ;4753 =00 CHECK.SBI.ERR[.C] ; Was there a time out on the write ?
U 10AD, 0000,003C,0180,0800,0000,10B0  ;4754  J/T.26.S3.NOTMO  ; No timeout
```

```
            ;4755  .PAGE
            ;4756 ;
            ;4757 ;////////////////////////////////////////////////////////////////////////////////
            ;4758 ;
            ;4759 ; Got time out ( SBI Error Bit <12> ) after write lock. Call an error.
            ;4760 ;
U 10AE. 0000.003D.5D80.0800.0084.70BE  ;4761  ERROR2[.7]
U 10AF. 0000.003C.0180.0800.0000.10B0  ;4762  NOP
            ;4763 ;
            ;4764 ;////////////////////////////////////////////////////////////////////////////////
            ;4765 ;
            ;4766 ; ERROR LOGOUT:
            ;4767 ;
            ;4768 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4769 ;  Modified Cycle Count
            ;4770 ;  NOT USED
            ;4771 ;  NOT USED
            ;4772 ;  Address Under Test
            ;4773 ;  Cycle Count for Single Stepping
            ;4774 ;  ID[sbi.err] Register
            ;4775 ;
            ;4776 ;////////////////////////////////////////////////////////////////////////////////
            ;4777 ;
            ;4778  ;
            ;4779  ; Check for any unexpected interrupts.
            ;4780  ;
            ;4781 =00
            ;4782 T.26.S3.NOTMO:
U 10B0. 0000.003D.0180.0800.0000.11F3  ;4783  CHECK.INTERRUPT  ; Check for interrupts
U 10B1. 0000.003C.0180.0800.0000.10B3  ;4784  J/T.26.S3.NOINT  ; No interrupt
```

```
            ;4785  .PAGE
            ;4786 ;
            ;4787 ;////////////////////////////////////////////////////////////////////////////
            ;4788 ;
            ;4789 ; Got an unexpected interrupt after lockwrite.  Call an error.
            ;4790 ;
U 10B2, 0000,003D,F580,0800,0084,70BE  ;4791  ERROR2[.A]
            ;4792 ;
            ;4793 ;////////////////////////////////////////////////////////////////////////////
            ;4794 ;
            ;4795 ; ERROR LOGOUT:
            ;4796 ;
            ;4797 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4798 ;  Modified Cycle Count
            ;4799 ;  NOT USED
            ;4800 ;  NOT USED
            ;4801 ;  Address Under Test
            ;4802 ;  Cycle Count for Single Stepping
            ;4803 ;  ID[vector] Register
            ;4804 ;  ID[sbi.err] Register
            ;4805 ;  ID[fault] Register
            ;4806 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4807 ;       0 = No Interrupt Occurred
            ;4808 ;
            ;4809 ;////////////////////////////////////////////////////////////////////////////
            ;4810 ;
            ;4811  ;
            ;4812  ; Check the data to insure that it is correct
            ;4813  ;
            ;4814 T.26.S3.NOINT:
U 10B3, 0010,0038,0180,0950,0200,12B4  ;4815  VA_RC[0A]  ; Load VA with test address
U 12B4, 0000,003C,0180,C800,0000,12B5  ;4816  D[LONG]_CACHE.P  ; Read data from LOCKWRITE
U 12B5, 0001,003C,0180,09D8,0000,12B6  ;4817  RC[0B]_D  ; Save in RC[0B]
U 12B6, 0019,0014,0580,0800,0010,12B7  ;4818  ALU_D+K[.1],CLK.UBCC ; Add 1 to read data - should get 0's
U 12B7, 0000,013C,0180,0800,0000,1194  ;4819  Z?
```

```
            ;4820   .PAGE
            ;4821 ;
            ;4822 ;////////////////////////////////////////////////////////////////////////////
            ;4823 ;
            ;4824 ; Got the wrong data when reading the data back.  Call an error.
            ;4825 ;
U 1194, 0000,003D,D580,0800,0084,70BE  ;4826 =0 ERROR2[.6]
            ;4827 ;
            ;4828 ;////////////////////////////////////////////////////////////////////////////
            ;4829 ;
            ;4830 ; ERROR LOGOUT:
            ;4831 ;
            ;4832 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4833 ;  Unused
            ;4834 ;  Expected Memory Data
            ;4835 ;  Received Memory Data
            ;4836 ;  Address Under Test
            ;4837 ;  Cycle Count for Single Stepping
            ;4838 ;
            ;4839 ;////////////////////////////////////////////////////////////////////////////
            ;4840 ;
U 1195, 0000,003C,0180,0800,0000,1196  ;4841  NOP
            ;4842 ;
            ;4843 ;============================================================================
            ;4844 ;
            ;4845 ; Subtest 4 Interlock Read
            ;4846 ;
U 1196, 0000,003D,0180,0800,0000,11CB  ;4847 =0 SUBTEST   ; Update the subtest counter
U 1197, 0000,003C,0180,0800,0000,1198  ;4848  NOP
U 1198, 0000,003D,0180,0800,0000,11C3  ;4849 =0 ERLOOP   ; Set the error loop
U 1199, 0000,003C,0180,0800,0000,119A  ;4850  NOP
U 119A, 0000,003D,0180,0800,0000,10CC  ;4851 =0 CALL[SBI.INIT]  ; Cleanup the sbi
U 119B, 0010,0038,0180,0950,0200,12B8  ;4852  VA_RC[0A]  ; Load VA with test address
U 12B8, 081B,001C,1980,0800,0000,12B9  ;4853  D_NOT.K[ZERO]  ; Load xFFFFFFFF in D to init the test location
U 12B9, 0001,003C,0180,09E0,0000,12BA  ;4854  RC[0C]_D  ; Save write data in RC[0C] in case error
U 12BA, 0000,003C,0180,A800,0000,119C  ;4855  CACHE.P_D[LONG]  ; Init to one's
U 119C, 0000,003D,8980,0800,0084,7206  ;4856 =0 SET.TRAP.MASK[.D] ; Enable timeout traps
U 119D, 0000,003C,0180,0800,0000,119E  ;4857  NOP
            ;4858 =0 D_RC[09],
U 119E, 0810,0039,0180,0948,0000,1251  ;4859      CALL[BUS]  ; Start single cycling
            ;4860  ;
            ;4861  ; Following sequence is performed in single cycle mode
            ;4862  ;
U 119F, 0000,003C,0180,E800,0000,12BB  ;4863  MCT/LOCKREAD.P  ; Perform a read first
U 12BB, 0001,003C,0180,09D8,0000,11A0  ;4864  RC[0B]_D  ; Save data from read in RC[0B]
            ;4865 =0 CALL[WAIT.LOOP],
U 11A0, 0810,0039,0180,0948,0010,1249  ;4866      D_RC[09],CLK.UBCC ; Wait in single cycle mode incase timeout
            ;4867  ;
            ;4868  ; End of single cycle mode
            ;4869  ;
U 11A1, 0000,003C,0180,0800,0000,10B4  ;4870  NOP
U 10B4, 0000,003D,0180,0800,0000,1200  ;4871 =00 CHECK.UTRAP  ; Was there a time out miro trap ?
U 10B5, 0000,003C,0180,0800,0000,10B8  ;4872  J/T.26.S4.NOTMO  ; No, check to see if any unexpected interrupts
```

```
            ;4873  .PAGE
            ;4874 ;
            ;4875 ;////////////////////////////////////////////////////////////////////////////////
            ;4876 ;
            ;4877 ; Got a time out micro trap on the lockread.  Call an error.
            ;4878 ;
U 10B6, 0000,003D,0180,0800,0084,70BE  ;4879  ERROR2[.8]
U 10B7, 0000,003C,0180,0800,0000,10B8  ;4880  NOP
            ;4881 ;
            ;4882 ;////////////////////////////////////////////////////////////////////////////////
            ;4883 ;
            ;4884 ; ERROR LOGOUT:
            ;4885 ;
            ;4886 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4887 ;  Unused
            ;4888 ;  Unused
            ;4889 ;  Unused
            ;4890 ;  Address Under Test
            ;4891 ;  Cycle Count for Single Stepping
            ;4892 ;  Expected Micro Trap Flags
            ;4893 ;  Received Micro Trap Flags
            ;4894 ;
            ;4895 ;////////////////////////////////////////////////////////////////////////////////
            ;4896 ;
            ;4897  ;
            ;4898  ; Check for any unexpected interrupts.
            ;4899  ;
            ;4900 =00
            ;4901 T.26.S4.NOTMO:
U 10B8, 0000,003D,0180,0800,0000,11F3  ;4902  CHECK.INTERRUPT  ; Check for interrupts
U 10B9, 0000,003C,0180,0800,0000,10BB  ;4903  J/T.26.S4.NOINT  ; No interrupt
```

```
            ;4904  .PAGE
            ;4905 ;
            ;4906 ;////////////////////////////////////////////////////////////////////////////////
            ;4907 ;
            ;4908 ; Got an unexpected interrupt on the lockread.  Call an error.
            ;4909 ;
U 10BA, 0000.003D.F580.0800.0084.70BE  ;4910  ERROR2[.A]
            ;4911 ;
            ;4912 ;////////////////////////////////////////////////////////////////////////////////
            ;4913 ;
            ;4914 ; ERROR LOGOUT:
            ;4915 ;
            ;4916 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4917 ;  Unused
            ;4918 ;  Unused
            ;4919 ;  Unused
            ;4920 ;  Address Under Test
            ;4921 ;  Unused
            ;4922 ;  ID[vector] Register
            ;4923 ;  ID[sbi.err] Register
            ;4924 ;  ID[fault] Register
            ;4925 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4926 ;       0 = No Interrupt Occurred
            ;4927 ;
            ;4928 ;////////////////////////////////////////////////////////////////////////////////
            ;4929 ;
            ;4930  ;
            ;4931  ; Check the data to insure that it is correct.
            ;4932  ;
            ;4933 T.26.S4.NOINT:
U 10BB, 0810.0038.0180.0958.0000.12BC  ;4934  D_RC[0B]  ; Load the D with the data from the lockread
U 12BC, 0019.0014.0580.0800.0010.12BD  ;4935  ALU_D+K[.1],CLK.UBCC ; Was data written ok?
U 12BD, 0000.013C.0180.0800.0000.11A2  ;4936  Z?   ; Branch if yes
```

```
            ;4937  .PAGE
            ;4938 ;
            ;4939 ;////////////////////////////////////////////////////////////////////////////////
            ;4940 ;
            ;4941 ; Got the wrong data from the lockread.  Call an error.
            ;4942 ;
U 11A2, 0000.003D.D580.0800.0084.70BE  ;4943 =0 ERROR2[.6]
            ;4944 ;
            ;4945 ;////////////////////////////////////////////////////////////////////////////////
            ;4946 ;
            ;4947 ; ERROR LOGOUT:
            ;4948 ;
            ;4949 ; DATA: I/O Base Address of MS780-E Currently under test
            ;4950 ;  Unused
            ;4951 ;  Expected Memory Data
            ;4952 ;  Received Memory Data
            ;4953 ;  Address Under Test
            ;4954 ;  Cycle Count for Single Stepping
            ;4955 ;
            ;4956 ;////////////////////////////////////////////////////////////////////////////////
            ;4957 ;
            ;4958  ;
            ;4959  ; Go back to the Restart entry point and test another MS780-E
            ;4960  ;
U 11A3, 0000.003C.0180.0800.0000.1098  ;4961  J/RESTART.TEST.26  ; Go see if more MS780-E's to test
            ;4962  ;
            ;4963  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4964  ;
            ;4965 T.26.EXIT:
U 12BE, 0000.003C.0180.0800.0000.11A4  ;4966  NOP
            ;4967
```

```
            ;4968  .PAGE "TEST 1B1 RESERVED"
            ;4969 =0
U 11A4, 0000,003D,0180,0800,0000,11AE  ;4970 T1B1: NEWTST
U 11A5, 0000,003C,0180,0800,0000,11A6  ;4971  NOP
            ;4972
```

```
            ;4973  .PAGE "TEST 1B2 RESERVED"
            ;4974 =0
U 11A6, 0000,003D,0180,0800,0000,11AE  ;4975 T1B2: NEWTST
U 11A7, 0000,003C,0180,0800,0000,12BF  ;4976  NOP
            ;4977
```

```
        ;4978  .PAGE "DUMMY NEWTST"
U 12BF, 0000,003D,0180,0800,0000,11AE  ;4979 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0857
ESKAR46.MCR                    MICRO2  1P(00)    26-JUL-85  17:14:55     Page  14
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2756= 2757= 2758= 2761=
U 1008  2410= 2412= 2414= 2419= 2260= 2261= 2262= 2263=
U 1010  2433= 2435= 2436= 2440= 2456= 2457= 2458= 2466=
U 1018  2468= 2470= 2472= 2473= 2399= 2400= 2401= 2402=
U 1020  2516= 2517= 2518= 2520= 2564= 2565= 2566= 2568=
U 1028  2883= 2884= 2885= 2886= 2887= 2888= 2889= 2890=
U 1030  3232= 3234= 3241= 3242= 3273= 3274= 3281= 3282=
U 1038  3303= 3304= 3311= 3335= 3390= 3391= 3398= 3399=
U 1040  3421= 3422= 3429= 3453= 3484= 3485= 3486= 3491=
U 1048  3573= 3574= 3575= 1902 1912= 1913= 2021: 1903
U 1050  3583= 3584= 3585= 3586= 3616= 3617= 3625= 3626=
U 1058  1965= 1967= 2067: 1904 1978= 1979= 2101: 1906
U 1060  3647= 3648= 3655= 3679= 3735= 3736= 3743= 3744=
U 1068  3766= 3767= 3774= 3798= 3855= 3856= 3863= 3864=
U 1070  3885= 3886= 3893= 3917= 3973= 3974= 3981= 3982=
U 1078  4004= 4005= 4012= 4036= 4130= 4132= 4139= 4140=
U 1080  4171= 4172= 4179= 4180= 4201= 4202= 4209= 4233=
U 1088  4288= 4289= 4296= 4297= 4319= 4320= 4327= 4351=
U 1090  4382= 4383= 4384= 4389= 4471= 4472= 4473= 1907
U 1098  4481= 4482= 4483= 4484= 4514= 4515= 4523= 4524=
U 10A0  4545= 4546= 4553= 4577= 4634= 4635= 4642= 4643=
U 10A8  4665= 4666= 4673= 4697= 4753= 4754= 4761= 4762=
U 10B0  4783= 4784= 4791= 4815= 4871= 4872= 4879= 4880=
U 10B8  4902= 4903= 4910= 4934= 2039= 2040= 2085= 2086=
U 10C0  2208= 2209= 2388= 2389= 2391= 2396= 2626= 2631=
U 10C8  2636= 2641= 2646= 2647= 2691= 2692= 2795= 2796=
U 10D0  2862= 2863= 2871= 2872= 2873= 2875= 2878= 2879=
U 10D8  2920= 2921= 2922= 2923= 2930= 2931= 2937= 2939=
U 10E0  2942= 2943= 2962= 2963= 2965= 2966= 3055= 3056=
U 10E8  3059= 3060= 3071= 3072= 3075= 3076= 3106= 3107=
U 10F0  3137= 3138= 3230= 3231= 3251= 3252= 3253= 3254=
U 10F8  3255= 3256= 3260= 3264= 3268= 3272= 3346= 3347=
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895: 1914 3367= 3368= 1896: 1897: 1898: 1899:
U 1110  3369= 3370= 3371= 3372= 3375= 3376= 3378= 3382=
U 1118  3385= 3389= 3462= 3463= 3571= 3572= 3594= 3595=
U 1120  3596= 3597= 3598= 3599= 3603= 3607= 3611= 3615=
U 1128  3690= 3691= 3711= 3712= 3713= 3714= 3715= 3716=
U 1130  3720= 3721= 3723= 3727= 3730= 3734= 3807= 3808=
U 1138  3829= 3830= 3831= 3832= 3833= 3834= 3840= 3844=
U 1140  3850= 3854= 3928= 3943= 3949= 3950= 3951= 3952=
U 1148  3953= 3954= 3958= 3959= 3961= 3965= 3968= 3972=
U 1150  4045= 4063= 4128= 4129= 4149= 4150= 4151= 4152=
U 1158  4153= 4154= 4158= 4162= 4166= 4170= 4244= 4245=
U 1160  4265= 4266= 4267= 4268= 4269= 4270= 4273= 4274=
U 1168  4276= 4280= 4283= 4287= 4360= 4361= 4469= 4470=
U 1170  4492= 4493= 4494= 4495= 4496= 4497= 4501= 4505=
U 1178  4509= 4513= 4588= 4589= 4610= 4611= 4612= 4613=
U 1180  4614= 4615= 4619= 4620= 4622= 4626= 4629= 4633=
U 1188  4706= 4721= 4727= 4728= 4729= 4730= 4731= 4732=
U 1190  4738= 4742= 4748= 4752= 4826= 4841= 4847= 4848=
U 1198  4849= 4850= 4851= 4852= 4856= 4857= 4859= 4863=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4866= 4870= 4943= 4961= 4970= 4971= 4975= 4976=
U 11A8  1915 1916 1917 1918 1919 1920 1954 1955
U 11B0  1956 1957 1958 1959 1960 1961 1962 1963
U 11B8  1964 1968 1969 1970 1971 1972 1973 1974
U 11C0  1975 1976 1977 1996 2019 2020 2065 2066
U 11C8  2149 2150 2151 2169 2171 2197 2198 2199
U 11D0  2200 2201 2202 2203 2204 2206 2207 2254
U 11D8  2255 2256 2258 2259 2397 2398 2423 2443
U 11E0  2444 2445 2452 2592 2593 2594 2595 2596
U 11E8  2597 2621 2669 2693 2694 2695 2696 2714
U 11F0  2715 2716 2717 2745 2747 2748 2749 2751
U 11F8  2752 2753 2754 2755 2762 2763 2764 2765
U 1200  2788 2789 2790 2791 2793 2794 2823 2824
U 1208  2825 2864 2865 2866 2867 2876 2877 2881
U 1210  2882 2895 2896 2898 2899 2900 2902 2907
U 1218  2908 2909 2911 2916 2918 2919 2927 2928
U 1220  2932 2933 2934 2935 2936 2944 2945 2946
U 1228  2948 2949 2950 2951 2952 2953 2958 2960
U 1230  2961 2964 2989 2990 2991 2992 3020 3021
U 1238  3023 3024 3025 3051 3053 3054 3058 3062
U 1240  3063 3064 3065 3067 3068 3069 3070 3073
U 1248  3074 3101 3132 3133 3135 3136 3153 3154
U 1250  3155 3170 3171 3172 3173 3243 3244 3245
U 1258  3257 3258 3259 3265 3266 3336 3337 3338
U 1260  3339 3373 3374 3377 3383 3454 3455 3587
U 1268  3588 3600 3601 3608 3609 3680 3681 3682
U 1270  3683 3717 3718 3719 3728 3799 3800 3823
U 1278  3835 3836 3837 3839 3846 3847 3848 3918
U 1280  3919 3920 3921 3955 3956 3957 3966 4037
U 1288  4038 4068 4141 4142 4143 4155 4156 4157
U 1290  4163 4164 4234 4235 4236 4237 4271 4272
U 1298  4275 4281 4352 4353 4485 4486 4498 4499
U 12A0  4506 4507 4578 4579 4580 4581 4616 4617
U 12A8  4618 4627 2127: 4698 4699 4733 4734 4735
U 12B0  4737 4744 4745 4746 4816 4817 4818 4819
U 12B8  4853 4854 4855 4864 4935 4936 4966 4979
U 12C0 - 13EF Unused
U 13F0  2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8  2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR46	720
Used	720
Remaining	

Total microwords used in memory U: 720
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR46.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 0861
ESKAR47.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1850 Micro Trap Handler and LSI-11 Support Routines
; 1851   Micro Trap Handler
; 1931 Micro Monitor Interface Routines
; 1932   Newtest Routine
; 1988   Error Loop Routine
; 2005   Error 1 Routine
; 2030   ERROR1 WITH PARAMETER
; 2051   Error 2 Routine
; 2077   ERROR 2 WITH PARAMETER
; 2096   Micro-Monitor Call
; 2110   Reserved Control Store
; 2137   Set Argument Count
; 2160   Increment Subtest Number
; 2180 Diagnostic Specific Subroutines
; 2181   Select Controller
; 2219   64K OR 256K CHIPS
; 2273   START TEST
; 2482   SELECT LOWER CONTROLLER
; 2530   SELECT UPPER CONTROLLER
; 2578   Set Diagnostic Mode Routine
; 2607   SBI Unjam Routine
; 2657   RESET SUBTEST NUMBER
; 2679   Initialize the SBI
; 2706   Disable Cache Routine
; 2727   Enable Cache
; 2748   Wait Loop Routine
; 2781   Check for Interrupt Routine
; 2828   CHECK UTRAP
; 2860   CHECK SBI.ERR
; 2890   Set Trap Mask
; 2918   FIND MS780-E MEMORY
; 3060   Generate IO Address
; 3085   Set Starting Address
; 3118   ENABLE NEXT MS780-E
; 3170   FIND CONTROLLERS
; 3252   GET MEMORY SIZE
; 3311   MS780-E/H CNFG REG CLEANUP
; 3382   INIT ROUTINE
; 3414   SETPSL ROUTINE
; 3428   CLRTIM ROUTINE
; 3442   CLRFLT ROUTINE
; 3456   CLRECC ROUTINE
; 3470   ENBECC ROUTINE
; 3484   ENBLFLT ROUTINE
; 3501   NOINTR ROUTINE
; 3544 TEST 1B3 SBI FUNCTIONAL CHECK OF VIRTUAL OPERATIONS
; 4176 TEST 1B4 INTERLEAVE MEMORY SIZING TEST
; 4790 TEST 1B5 RESERVED
; 4795 TEST 1B6 RESERVED

```
: 4800 DUMMY NEWTST
```

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806  .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR47
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ; MODIFIED BY:
;1836  ;
;1837  ;  1.1 March 10, 1983  Paul Hakala
;1838  ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839  ;   the correct memory address when 256k ram chips are
;1840  ;   present.
;1841  ;
;1842  ;
;1843  ;
;1844  ;
;1845  ;*****************************************************************************
;1846  ;
;1847  ;
```

```
;1848  .PAGE
;1849
;1850  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1851  .TOC "   Micro Trap Handler"
;1852 ;++
;1853 ; FUNCTIONAL DESCRIPTION:
;1854 ;
;1855 ; This routine is responsible for 'catching' micro-traps
;1856 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1857 ; contains the Trap-Expected Mask. If the specified bit is set in
;1858 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1859 ; an error call is made.
;1860 ;
;1861 ; INPUT PARAMETERS:
;1862 ;
;1863 ; R[0F] - Expected Trap Mask
;1864 ;  Trap Code Function
;1865 ;  ---------     --------
;1866 ;       0  System Init
;1867 ;       1  Data Unaligned
;1868 ;       2  Cross Page
;1869 ;       3  TB Modify
;1870 ;       4  TB Protection
;1871 ;       6  TB Miss
;1872 ;       7  TB Parity
;1873 ;       8  Cache Parity
;1874 ;       9  Reserved Floating Operand
;1875 ;       C  Read Data Substitute
;1876 ;       D  Time out
;1877 ;       F  Control Store Parity Error
;1878 ;      10  Odd Address
;1879 ;
;1880 ; OUTPUT PARAMETERS:
;1881 ;
;1882 ; R[0F] - Mask bit is cleared.
;1883 ;
;1884 ; DATA: Micro PC of Micro Trap
;1885 ;  Trap Code (See Above)
;1886 ;  Fault Status Register
;1887 ;  SBI Error Register
;1888 ;  Timeout Address Register
;1889 ;  Maintenance Register
;1890 ;  Cache Parity Register
;1891 ;  TB Error Register 1
;1892 ;  TB Error Register 0
;1893 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1894 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1895 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1896 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1897 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1898 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1899 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1900 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1901 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1902 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000,003C,8580,0800,0084,7001  ;1903 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000,003C,8980,0800,0084,7001  ;1904 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000,003C,6580,0800,0084,7001  ;1905 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000,003C,6180,0800,0084,7003  ;1906 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1907
U 1001, 0000,003C,81F0,2C00,0000,1036  ;1908 trph0: q_id[ustack]  ; Get upc of trap
U 1036, 0C00,003C,01E0,0800,0000,103E  ;1909  q_d,d_q   ; Setup to put it back on stack
U 103E, 0C00,003C,81E0,3C00,0000,1046  ;1910  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1046, 0000,003C,01C0,0A78,0000,1056  ;1911  q_r[0f]   ; Get the Expected Trap Mask
        ;1912  alu_q.andnot.mask,
U 1056, 0001,2024,0180,0800,0010,105B  ;1913   clk.ubcc ; Was trap expected?
U 105B, 0000,013C,0180,0800,0000,1002  ;1914  z?
        ;1915 =0 r[0f]_alu,  ; It was, clear the mask and return
        ;1916      alu_q.and.mask, ; ...
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1917       return0  ; ...
U 1003, 0018,0038,1D80,09E8,0000,1014  ;1918 trph1: rc[0d]_sc  ; Output the Trap Code
U 1014, 0000,003D,D980,0800,0084,7202  ;1919 =0 setrcf[.9]  ; Set output count
U 1015, 0000,003C,49F0,2C00,0000,1066  ;1920  q_id[tber0]  ; Output all the error registers
U 1066, 0001,203C,4DF0,2DB0,0000,106B  ;1921  rc[6]_q,q_id[tber1] ; ...
U 106B, 0001,203C,65F0,2DB8,0000,106E  ;1922  rc[7]_q,q_id[sbi.err] ; ...
U 106E, 0001,203C,6DF0,2DD8,0000,1073  ;1923  rc[0b]_q,q_id[fault] ; ...
U 1073, 0001,203C,75F0,2DE0,0000,1077  ;1924  rc[0c]_q,q_id[maint] ; ...
U 1077, 0001,203C,79F0,2DC8,0000,107B  ;1925  rc[9]_q,q_id[parity] ; ...
U 107B, 0001,203C,69F0,2DC0,0000,1085  ;1926  rc[8]_q,q_id[time.addr] ; ...
U 1085, 0001,203C,81F0,2DD0,0000,1000  ;1927  rc[0a]_q,q_id[ustack] ; ...
        ;1928 1000: ERROR1[.C],  ;
U 1000, 0001,203D,8580,09F0,0084,703C  ;1929   rc[0e]_q ; Report the error
```

```
            ;1930  .PAGE
            ;1931  .TOC " Micro Monitor Interface Routines"
            ;1932  .TOC "   Newtest Routine"
            ;1933 ;++
            ;1934 ; FUNCTIONAL DESCRIPTION:
            ;1935 ;
            ;1936 ; This routine Initializes the following registers:
            ;1937 ;  PSL    to 1F
            ;1938 ;  CES    to 0
            ;1939 ;  ACC.CS    to disable accellerator
            ;1940 ;  SIR    to 0
            ;1941 ;  CLK.CS    to stop interval timer
            ;1942 ;  TBER0    to disable the TB
            ;1943 ;  SBI.MAINT to     0
            ;1944 ;  SBI.ERR    to 0
            ;1945 ;  FAULT    to 0
            ;1946 ;  COMP    to 0
            ;1947 ;  RC[0E]    to  0
            ;1948 ;  R[0F]    to 0
            ;1949 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1950 ; A SCOPE call is then made to the Monitor.
            ;1951 ;
            ;1952 ; CALLING CONSTRAINT:
            ;1953 ;
            ;1954 ; =0
            ;1955 ;
            ;1956 ; SIDE EFFECTS:
            ;1957 ;
            ;1958 ; D Reg is destroyed
            ;1959 ; SC is destroyed
            ;1960 ;--
U 108F, 0000.003C,65F8,0800,0084,7093  ;1961 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1093, 0818.0038,8D80,0800,0000,1097  ;1962  d_k[.1f]  ; D=1F
U 1097, 0D00.003C,0180,0800,0000,109B  ;1963  d_dal.sc  ; D=001F0000
U 109B, 0000.003C,3D80,3C00,0000,109F  ;1964  id[psl]_d  ; psl = 001F0000
U 109F, 0818.0038,0580,0800,0000,10A3  ;1965  d_k[.1]  ; Clear the CES register
U 10A3, 0D00.003C,0180,0800,0000,10A7  ;1966  d_dal.sc  ; D = 00010000
U 10A7, 0F00.003C,3180,3C00,0000,10AB  ;1967  id[ces]_d,d_0  ; ces = 00010000
U 10AB, 0000.003C,5D80,3C00,0000,10AF  ;1968  id[acc.cs]_d  ; acc.cs = 0
U 10AF, 0000.003C,3980,3C00,0000,10B3  ;1969  id[sir]_d  ; sir = 0
U 10B3, 0000.003C,2980,3C00,0000,10B7  ;1970  id[clk.cs]_d  ; clk.cs = 0
U 10B7, 0000.003C,4980,3C00,0000,1024  ;1971  id[tber0]_d  ; tber0 = 0
U 1024, 0000.003D,0180,0800,0000,1223  ;1972 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1973  intrpt.strobe,  ; Strobe interrupts
U 1025, 0818.0038,C180,0800,4000,10BB  ;1974      d_k[.ffff]  ; Setup to clear SBI error register and
U 10BB, 0801.0028,6580,3C00,0000,10BF  ;1975  id[sbi.err]_d,d_not.d ; and fault register
U 10BF, 0F00.003C,6D80,3C00,0000,10CF  ;1976  id[fault]_d,d_0  ; ...
U 10CF, 0000.003C,6D80,3C00,0000,10D3  ;1977  id[fault]_d  ; fault = 0
U 10D3, 0000.003C,7580,3C00,0000,10D7  ;1978  id[maint]_d  ; MAINT = 0
U 10D7, 0000.003C,6580,3C00,0000,10E3  ;1979  id[sbi.err]_d  ; sbi error reg = 0
U 10E3, 0000.003C,7180,3C00,0000,1109  ;1980  id[comp]_d  ; comp reg = 0
U 1109, 0000.003C,E580,3C00,0000,11FA  ;1981  ID[39]_d  ; Clear code for subroutine START.TEST
U 11FA, 0001.003C,0180,09F0,0000,11FB  ;1982  rc[0e]_d  ;
U 11FB, 0001.003C,0180,0AF8,0000,11FC  ;1983  r[0f]_d  ; Clear expected trap mask
U 11FC, 0001.003C,0180,09F8,0000,1034  ;1984  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1034, 0000,003D,0180,0800,0000,1229  ;1985 =0 call[disable.cache] ; Disable the cache
U 1035, 0F03,003C,0180,09E8,0000,105E  ;1986  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1987   .PAGE
            ;1988   .TOC "    Error Loop Routine"
            ;1989 ;++
            ;1990 ; FUNCTIONAL DESCRIPTION:
            ;1991 ;
            ;1992 ; This routine is called with the "ERLOOP" macro. The
            ;1993 ; routine calls the Micro Monitor to setup an error loop.
            ;1994 ;
            ;1995 ; CALLING CONSTRAINT:
            ;1996 ;
            ;1997 ; =0
            ;1998 ;
            ;1999 ; SIDE EFFECTS:
            ;2000 ;
            ;2001 ; D Reg - Destroyed
            ;2002 ;--
U 11FD, 0818,0038,0980,0800,0000,105E  ;2003 ERLOOP: d_k[.2],j/callcon
```

```
            ;2004  .PAGE
            ;2005  .TOC "   Error 1 Routine"
            ;2006 ;++
            ;2007 ; FUNCTIONAL DESCRIPTION:
            ;2008 ;
            ;2009 ; This routine is used to report an error to the user. This
            ;2010 ; routine is used when you do not wish to continue in the
            ;2011 ; same test after the call to the Monitor.
            ;2012 ;
            ;2013 ; CALLING CONSTRAINT:
            ;2014 ;
            ;2015 ; None
            ;2016 ;
            ;2017 ; INPUTS:
            ;2018 ;
            ;2019 ; rc[0f]<15:8> - Contain the subtest number
            ;2020 ;
            ;2021 ; OUTPUTS:
            ;2022 ;
            ;2023 ; D<15:8> - Subtest number
            ;2024 ; D<7:0> - Error 1 Monitor Code
            ;2025 ;--
U 11FE, 0810.0038.0180.0978.0000.11FF  ;2026 ERROR1: d_rc[0f]  ; Get the subtest number
U 11FF, 0819.0034.4D80.0800.0000.104E  ;2027  d_d.and.k[.ff00] ; ...
U 104E, 0819.0030.1180.0800.0000.105E  ;2028 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2029  .PAGE
            ;2030  .TOC "    ERROR1 WITH PARAMETER"
            ;2031 ;++
            ;2032 ; FUNCTIONAL DESCRIPTION:
            ;2033 ;
            ;2034 ; This subroutine takes  as parameter the number of arguments
            ;2035 ; to be printed to the console in the case of an error logout.
            ;2036 ;
            ;2037 ; CALLING CONSTRAINT:
            ;2038 ;
            ;2039 ; =0
            ;2040 ;INPUTS:
            ;2041 ;
            ;2042 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2043 ;--
            ;2044 =0
            ;2045 ERR1.ARG:
U 103C, 0000,003D,0180,0800,0000,1202  ;2046  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 103D, 0000,003C,0180,0800,0000,11FE  ;2047  J/ERROR1  ; Go to ERROR1
            ;2048 ;
            ;2049 ;
```

```
            ;2050  .PAGE
            ;2051  .TOC "   Error 2 Routine"
            ;2052 ;**
            ;2053 ; FUNCTIONAL DESCRIPTION:
            ;2054 ;
            ;2055 ; This routine is used  to report an error to the user. This
            ;2056 ; routine is used when you wish to continue in the same test
            ;2057 ;       after the call to the Monitor.
            ;2058 ;
            ;2059 ; CALLING CONSTRAINT:
            ;2060 ;
            ;2061 ; =0
            ;2062 ;
            ;2063 ; INPUTS:
            ;2064 ;
            ;2065 ; rc[0f]<15:8> - Contain the subtest number
            ;2066 ;
            ;2067 ; OUTPUTS:
            ;2068 ;
            ;2069 ; D<15:8> - Subtest number
            ;2070 ; D<7:0> - Error 2 Monitor Code
            ;2071 ;--
U 1200,  0810.0038.0180.0978.0000.1201  ;2072 ERROR2: d_rc[0f]  ; Get the subtest number
U 1201,  0819.0034.4D80.0800.0000.105A  ;2073  d_d.and.k[.ff00] ; ...
U 105A,  0819.0030.D580.0800.0000.105E  ;2074 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2075 ;
```

```
            ;2076  .PAGE
            ;2077  .TOC "   ERROR 2 WITH PARAMETER"
            ;2078 ;++
            ;2079 ; FUNCTIONAL DESCRIPTION:
            ;2080 ;
            ;2081 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2082 ; except that in this case after setting the number of arguments
            ;2083 ; too the console, control is transferred to routine ERROR2.
            ;2084 ;
            ;2085 ; INPUTS:
            ;2086 ;
            ;2087 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2088 ;
            ;2089 ;--
            ;2090 =0
            ;2091 ERR2.ARG:
U 1044, 0000,003D,0180,0800,0000,1202  ;2092  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1045, 0000,003C,0180,0800,0000,1200  ;2093  J/ERROR2  ; Go to ERROR2
            ;2094 ;
```

```
            ;2095  .PAGE
            ;2096  .TOC "   Micro-Monitor Call"
            ;2097 ;*
            ;2098 ; FUNCTIONAL DESCRIPTION:
            ;2099 ;
            ;2100 ; This micro word calls the micro monitor.
            ;2101 ;
            ;2102 ; EXPLICIT INPUTS:
            ;2103 ;
            ;2104 ; D reg must contain valid monitor code.
            ;2105 ;--
            ;2106 105E:
            ;2107 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2108  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2109  .PAGE
            ;2110  .TOC "   Reserved Control Store"
            ;2111 ;++
            ;2112 ; FUNCTIONAL DESCRIPTION:
            ;2113 ;
            ;2114 ; The following 16 locations must be reserved so the monitor
            ;2115 ; can use them to perform various functions that require
            ;2116 ; the execution of micro-code.
            ;2117 ;--
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2118 13F0: nop
U 13F1, 0000.003C.0180.0800.0000.13F2  ;2119 13F1: nop
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2120 13F2: nop
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2121 13F3: nop
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2122 13F4: nop
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2123 13F5: nop
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2124 13F6: nop
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2125 13F7: nop
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2126 13F8: nop
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2127 13F9: nop
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2128 13FA: nop
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2129 13FB: nop
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2130 13FC: nop
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2131 13FD: nop
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2132 13FE: nop
U 13FF, 0000.003C.0180.0800.0000.12AA  ;2133 13FF: nop
U 12AA, 0000.003C.0180.0800.0000.1202  ;2134 12AA: nop ; This location is permanently blasted in the FPLA
            ;2135   ; to cause a micro ECO to location 12AA.
```

```
            ;2136  .PAGE
            ;2137  .TOC "   Set Argument Count"
            ;2138 ;++
            ;2139 ; FUNCTIONAL DESCRIPTION:
            ;2140 ;
            ;2141 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2142 ; to the console.
            ;2143 ;
            ;2144 ; CALLING CONSTRAINT:
            ;2145 ;
            ;2146 ; =0
            ;2147 ;
            ;2148 ; INPUTS:
            ;2149 ;
            ;2150 ; SC - Contains new value of argument count
            ;2151 ;
            ;2152 ; SIDE EFFECTS:
            ;2153 ;
            ;2154 ; Q is destroyed
            ;2155 ;--
U 1202, 0010,0038,01C0,0978,0000,1203  ;2156 SETRCF: q_rc[0f]  ; Get the argument count
U 1203, 0019,2034,4DC0,0800,0000,1204  ;2157  q_q.and.k[.ff00] ; ...
U 1204, 0019,2032,1D80,09F8,0000,0001  ;2158  rc[0f]_q.or.sc,returni ; Set new argument count and return
```

```
            ;2159  .PAGE
            ;2160  .TOC "    Increment Subtest Number"
            ;2161 ;++
            ;2162 ; FUNCTIONAL DESCRIPTION:
            ;2163 ;
            ;2164 ; This routine is used to increment the subtest number
            ;2165 ; in RC[0F]<15:8>.
            ;2166 ;
            ;2167 ; CALLING CONSTRAINT:
            ;2168 ;
            ;2169 ; =0
            ;2170 ;
            ;2171 ; SIDE EFFECTS:
            ;2172 ;
            ;2173 ; D register is destroyed
            ;2174 ;--
            ;2175 subtst:
U 1205, 0810,0038,4D80,0978,0000,1206  ;2176  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2177  rc[0f]_d+k[.ff00], ; Increment and return
U 1206, 0019,4016,4D80,09F8,0000,0001  ;2178      word,return1 ; (Subtest number is negative)
```

```
            ;2179  .PAGE
            ;2180  .TOC " Diagnostic Specific Subroutines"
            ;2181  .TOC "   Select Controller"
            ;2182 ;++
            ;2183 ; FUNCTIONAL DESCRIPTION:
            ;2184 ;
            ;2185 ; This routine is used to put the memory system into the
            ;2186 ; specified configuration with respect to controller select.
            ;2187 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2188 ; a RETURN2 is made.
            ;2189 ;
            ;2190 ; CALLING CONSTRAINT:
            ;2191 ;
            ;2192 ; =00
            ;2193 ;
            ;2194 ; INPUTS:
            ;2195 ;
            ;2196 ; d - Contains value to write to bits<2:0> of Register A
            ;2197 ; rc[0e] - Address of Register A
            ;2198 ;
            ;2199 ; SIDE EFFECTS:
            ;2200 ;
            ;2201 ; sc,d,va are destroyed
            ;2202 ;--
            ;2203 SELECT.CNTRLR:
U 1207, 0010,0038,0180,0970,0284,7208  ;2204  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1208, 0801,001C,0180,0800,0000,1209  ;2205  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1209, 0000,003C,0180,A800,0000,120A  ;2206  cache.p_d[long]  ; Write the configuration Register
U 120A, 0818,0038,5D80,0800,0000,120B  ;2207  d_k[.7]  ; Get Misconfiguration bits
U 120B, 0000,003C,61F8,0800,0084,720C  ;2208  sc_k[.F],q_0  ; ...
U 120C, 0D00,003C,0180,0800,0000,120D  ;2209  d_dal.sc  ; ... D = x38000
U 120D, 0000,003C,01E0,0800,0000,120E  ;2210  q_d  ; Save mask
U 120E, 0000,003C,0180,C800,0000,120F  ;2211  d[long]_cache.p  ; Read CNFG A Reg and
            ;2212  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 120F, 001D,2034,0180,0800,0010,1210  ;2213  clk.ubcc ; ...
U 1210, 0000,013C,0180,0800,0000,104C  ;2214  z?  ; Branch if no
U 104C, 0000,003E,0180,0800,0000,0001  ;2215 =0 RETURN1  ; Bad configuration
U 104D, 0000,003E,0180,0800,0000,0002  ;2216  RETURN2  ; Configuration ok
            ;2217
```

```
;2218  .PAGE
;2219  .TOC "   64K OR 256K CHIPS"
;2220 ;*************************************************************
;2221 ;++
;2222 ;
;2223 ; Functional Description:
;2224 ; ----------------------
;2225 ; This subroutine is used to find the type of chips
;2226 ; on the MS780-E/H arrays.  The RETURN modes are as
;2227 ; follows:
;2228 ;
;2229 ; RETURN1 = Error. Mixed 64K and 256K arrays
;2230 ;
;2231 ; RETURN2 = 64K chips on the array
;2232 ;
;2233 ; RETURN3 = 256K chips on the array
;2234 ;
;2235 ; Calling Constraint: =00
;2236 ; ------------------
;2237 ;
;2238 ;
;2239 ; Inputs:
;2240 ; ------
;2241 ;
;2242 ; RC[0E] must hold the I/O base address of the MS780-E/H
;2243 ; currently under test
;2244 ;
;2245 ; Outputs:
;2246 ; -------
;2247 ;
;2248 ; NO specific outputs. (See RETURN modes above)
;2249 ;
;2250 ;
;2251 ; Side Effects:
;2252 ; ------------
;2253 ;
;2254 ; The contents of the following registers will be lost:
;2255 ;
;2256 ; D
;2257 ;
;2258 ;--
;2259 ;*************************************************************
;2260 64K.OR.256K:
U 1211, 0010,0038,0180,0970,0200,1212  ;2261  VA_RC[0E]    ; Point to CNFG Register A
U 1212, 0000,003C,0180,C800,0000,1213  ;2262  D[LONG]_CACHE.P    ; Read the register
U 1213, 0200,003C,0180,0800,0000,1214  ;2263  D_D.RIGHT2    ; Shift received contents two bits
        ;2264        ; ...to the right
U 1214, 0600,003C,0180,0800,0000,1215  ;2265  D_D.RIGHT    ; Shift D one more time
U 1215, 0000,193C,0180,0800,0000,100C  ;2266  D1-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2267 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2268  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2269  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2270  RETURN1     ; Error: missing arrays
        ;2271
```

```
;2272  .PAGE
;2273  .TOC "   START TEST"
;2274 ;*************************************************************
;2275 ;**
;2276 ;
;2277 ; Functional Description:
;2278 ; ---------------------
;2279 ;
;2280 ;         This subroutine will be called by all tests that  check  the
;2281 ;    controllers,  or  those  tests  that  have  to switch between the
;2282 ;    controllers when executing.   Its  main  functions  are:   select
;2283 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2284 ;    test that called it, call out an error  if  neither  one  of  the
;2285 ;    controllers can be configured properly, if all the controllers on
;2286 ;    a MS780-E/H were configured and tested, it should select the next
;2287 ;    MS780-E/H on the SBI and repeat the above sequence.
;2288 ;    A test making use of this  subroutine  should  be  configured  as
;2289 ;    follows:
;2290 ;
;2291 ;
;2292 ;        Begin TEST.XX
;2293 ;        :
;2294 ;        :   Call NEWTST
;2295 ;        :   Call FIND.MS780E
;2296 ;        :   IF No MS780-E's on the SBI
;2297 ;        :     THEN
;2298 ;        :          Skip to End of TEST.XX
;2299 ;        :     ELSE
;2300 ;        :
;2301 ;        : RESTART.TEST.XX:
;2302 ;        :
;2303 ;        :   Call START TEST
;2304 ;        :   IF Return1 from START.TEST
;2305 ;        :     THEN
;2306 ;        :          Skip to End of TEST.XX
;2307 ;        :     ELSE.
;2308 ;        :
;2309 ;        ~
;2310 ;
;2311 ;        :
;2312 ;        :   Body of TEST.XX
;2313 ;        :
;2314 ;        ~
;2315 ;
;2316 ;        :   Jump RESTART.TEST.XX
;2317 ;        :
;2318 ;        End TEST.XX
;2319 ;
;2320 ;
;2321 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2322 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2323 ;    should be passed when the subroutine is called a second, third or
;2324 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2325 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2326 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2327 ;    Register 39 can be interpreted as follows:
;2328 ;
;2329 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2330 ;               called the first time.  When this code is encountered
;2331 ;               this subroutine will try to select the lower controller.
;2332 ;               If the lower controller is misconfigured, the pointer in
;2333 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2334 ;               is re-entered.
;2335 ;               If, however there is no misconfiguration, the value in
;2336 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2337 ;               calling test.
;2338 ;
;2339 ;     b01   -   This value signifies that an attempt at configuring the
;2340 ;               lower controller failed, and the subroutine will attempt
;2341 ;               to select the upper controller.
;2342 ;               If the upper controller is also misconfigured execution
;2343 ;               stops with a call to ERROR1.
;2344 ;               If there is no misconfiguration on this controller, then
;2345 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2346 ;               made to the calling test.
;2347 ;
;2348 ;     b10   -   This value signifies that the lower controller was selected
;2349 ;               previously and the test was executed once.  If the
;2350 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2351 ;               39 is loaded with b11 and an attempt is made to select the
;2352 ;               upper controller.
;2353 ;               If there is a misconfiguration on this controller, this
;2354 ;               subroutine is just re-entered.
;2355 ;               Otherwise, a Return2 will be made to the calling test.
;2356 ;
;2357 ;     b11   -   This code identifies the cases in which the test has
;2358 ;               been executed at least once (i.e., at least one of the
;2359 ;               controllers on the MS780-E unit under test could be
;2360 ;               configured properly).  When this code is detected the
;2361 ;               subroutine clears ID Reg 39 and makes a call to
;2362 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2363 ;               tested then the subroutine executes a Return1.
;2364 ;               Otherwise the subroutine is re-entered.
;2365 ;
;2366 ;
;2367 ; Calling Constraint: =00
;2368 ; ------------------
;2369 ;
;2370 ;
;2371 ; Inputs:
;2372 ; ------
;2373 ;
;2374 ; ID Register 39 must be cleared by NEWTST
;2375 ;
;2376 ;
;2377 ; Outputs:
;2378 ; -------
;2379 ;
;2380 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2381 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2382 ;         to be tested
            ;2383 ;
            ;2384 ; Side Effects:
            ;2385 ; ------------
            ;2386 ;
            ;2387 ; Contents of the following regiters will be destroyed:
            ;2388 ;
            ;2389 ; D, Q
            ;2390 ;
            ;2391 ;--
            ;2392 ;*************************************************************
            ;2393 =0
            ;2394 START.TEST:
U 1054, 0000,003D,0180,0800,0000,1224  ;2395  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 1055, 0000,003C,0180,0800,0000,1058  ;2396  NOP    ; ...tests that have more than one
            ;2397 =0     ; ...subtest.)
U 1058, 0000,003D,0180,0800,0000,11FD  ;2398  ERLOOP    ; Set up the error loop for the
            ;2399       ; ...eventuality that the MS780-E/H
            ;2400       ; ...currently under test has no
            ;2401       ; ...Controllers
            ;2402 RE.ENTRY:    ; Re entry point for this routine
U 1059, 0000,003C,E5F0,2C00,0000,1216  ;2403  Q_ID[39]    ; Get the code
U 1216, 0C00,003C,0180,0800,0000,1217  ;2404  D_Q    ; Put it in the D Register
U 1217, 0000,193C,0180,0800,0000,101C  ;2405  DI-0?    ; Branch on the code
U 101C, 0000,003C,0180,0800,0000,1008  ;2406 =1100 J/STRT.CASE00    ; Code is 00
U 101D, 0000,003C,0180,0800,0000,1010  ;2407  J/STRT.CASE01    ; Code is 01
U 101E, 0000,003C,0180,0800,0000,121C  ;2408  J/STRT.CASE10    ; Code is 10
U 101F, 0000,003C,0180,0800,0000,101B  ;2409  J/STRT.CASE11    ; Code is 11
            ;2410 ;
            ;2411 ;
            ;2412 ; At this point the code in ID[39] was 00
            ;2413 ;
            ;2414 ;
            ;2415 =00
            ;2416 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1030  ;2417  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2418  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,1218  ;2419   D_K[.1]    ; Set up the code for re entry to this
            ;2420       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2421  D_K[.2]    ; Set up the code for a next call to
            ;2422       ; ...START.TEST indicating that the
            ;2423       ; ...Lower Controller was configured
            ;2424       ; ...( 10 )
            ;2425  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2426   RETURN2    ; Let the test know that the Lower
            ;2427       ; ...has been configured
            ;2428 STRT.LOW.MIS:
            ;2429  ID[39]_D,    ; Save code in ID[39] signifing that
U 1218, 0000,003C,E580,3C00,0000,1059  ;2430   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2431       ; ...and re enter this subroutine
            ;2432
            ;2433 ;
            ;2434 ;
            ;2435 ; At this point the code is 01
            ;2436 ;
```

```
            ;2437 ;
            ;2438 =00
            ;2439 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1038  ;2440  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2441  J/STRT.UPR.MIS.   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,1219  ;2442   D_0   ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2443  D_K[.3]    ; Upper controller was configured
            ;2444       ; ...thus the code must be changed
            ;2445       ; ...accordingly ( 11 )
            ;2446  ID[39]_D,   ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2447   RETURN2   ; Let the test know that the Upper
            ;2448       ; ...Controller has been configured
            ;2449 STRT.UPR.MIS:
U 1219, 0000,003C,E580,3C00,0000,121A  ;2450  ID[39]_D   ; Save the Code ( 00 )
U 121A, 0018,0038,0D80,09E8,0000,121B  ;2451  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 121B, 0000,003D,0980,0800,0084,703C  ;2452  ERROR1[.2]   ; Flag the error.
            ;2453 ;
            ;2454 ;
            ;2455 ; At this point the code is 10
            ;2456 ;
            ;2457 ;
            ;2458 STRT.CASE10:
U 121C, 0818,0038,0D80,0800,0000,1018  ;2459  D_K[.3]    ; Set the code to 11
            ;2460
            ;2461
            ;2462 =00 ID[39]_D,   ; Save the code
U 1018, 0000,003D,E580,3C00,0000,1038  ;2463   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1019, 0000,003C,0180,0800,0000,1059  ;2464  J/RE.ENTRY   ; Misconfigured Upper Controller
U 101A, 0000,003E,0180,0800,0000,0002  ;2465  RETURN2    ; Upper Controller configured
            ;2466       ; ...let the test know it
            ;2467 ;
            ;2468 ;
            ;2469 ; At this point the code is 11
            ;2470 ;
            ;2471 ;
            ;2472 STRT.CASE11:
U 101B, 0F00,003C,0180,0800,0000,1020  ;2473  D_0    ; Clear the code
            ;2474 =00 ID[39]_D,   ; Save the code
U 1020, 0000,003D,E580,3C00,0000,127E  ;2475   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2476       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,1059  ;2477  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2478       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2479  RETURN1    ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1030  ;2480  NOP
```

```
            ;2481   .PAGE
            ;2482   .TOC "   SELECT LOWER CONTROLLER"
            ;2483 ;*****************************************************************
            ;2484 ;++
            ;2485 ;
            ;2486 ; Functional Description:
            ;2487 ; ----------------------
            ;2488 ;
            ;2489 ; This subroutine can be called to select the lower
            ;2490 ; Controller on a MS780-E/H.
            ;2491 ; If the Lower controller is misconfigured then a
            ;2492 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2493 ;
            ;2494 ; Calling Constraint: =00
            ;2495 ; ------------------
            ;2496 ;
            ;2497 ;
            ;2498 ; Inputs:
            ;2499 ; ------
            ;2500 ;
            ;2501 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2502 ;
            ;2503 ; Outputs:
            ;2504 ; -------
            ;2505 ;
            ;2506 ; RC[0D] will have the address of CNFG Register C
            ;2507 ;        ( 2000x008 )
            ;2508 ;
            ;2509 ; Side Effects:
            ;2510 ; ------------
            ;2511 ;
            ;2512 ; Contents of the following registers will lost:
            ;2513 ;
            ;2514 ; D
            ;2515 ;
            ;2516 ; The Lower controller on the MS780-E/H will be configured
            ;2517 ; when the subroutine is exitted with a RETURN2
            ;2518 ;--
            ;2519 ;*****************************************************************
            ;2520 =00
            ;2521 SELECT.LOWER:
            ;2522   D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1030, 0818,0039,1980,0800,0000,1207  ;2523    CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1031, 0000,003E,0180,0800,0000,0001  ;2524  RETURN1     ; Lower Controller Misconfigured
U 1032, 0810,0038,0180,0970,0000,1033  ;2525  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2526   RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1033, 0019,0016,0180,09E8,0000,0002  ;2527    RETURN2    ; Let caller know that the controller
            ;2528        ; ...was configured properly
```

```
;2529  .PAGE
;2530  .TOC "   SELECT UPPER CONTROLLER"
;2531 ;*************************************************************
;2532 ;++
;2533 ;
;2534 ; Functional Description:
;2535 ; ----------------------
;2536 ;
;2537 ; This subroutine can be called to select the upper
;2538 ; Controller on a MS780-E/H.
;2539 ; If the Upper controller is misconfigured then a
;2540 ; RETURN1 will be made.  Otherwise a RETURN2
;2541 ;
;2542 ; Calling Constraint: =00
;2543 ; -----------------
;2544 ;
;2545 ;
;2546 ; Inputs:
;2547 ; ------
;2548 ;
;2549 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2550 ;
;2551 ; Outputs:
;2552 ; -------
;2553 ;
;2554 ; RC[0D] will have the address of CNFG Register D
;2555 ;        ( 2000x010 )
;2556 ;
;2557 ; Side Effects:
;2558 ; ------------
;2559 ;
;2560 ; Contents of the following registers will lost:
;2561 ;
;2562 ; D
;2563 ;
;2564 ; The Upper controller on the MS780-E/H will be configured
;2565 ; when the subroutine is exitted with a RETURN2
;2566 ;--
;2567 ;*************************************************************
;2568 =00
;2569 SELECT.UPPER:
;2570  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1038, 0818,0039,0980,0800,0000,1207  ;2571   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1039, 0000,003E,0180,0800,0000,0001  ;2572  RETURN1    ; Upper Controller Misconfigured
U 103A, 0810,0038,0180,0970,0000,103B  ;2573  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2574  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 103B, 0019,0016,8580,09E8,0000,0002  ;2575   RETURN2   ; Let caller know that the controller
;2576       ; ...was configured properly
```

```
            ;2577  .PAGE
            ;2578  .TOC "   Set Diagnostic Mode Routine"
            ;2579 ;++
            ;2580 ; FUNCTIONAL DESCRIPTION:
            ;2581 ;
            ;2582 ; This routine is used to set the specified diagnostic mode
            ;2583 ; in Register B. Other bits in the register remain unchanged.
            ;2584 ;
            ;2585 ; CALLING CONSTRAINT:
            ;2586 ;
            ;2587 ; =0
            ;2588 ;
            ;2589 ; INPUTS:
            ;2590 ;
            ;2591 ; d - Contains the mode to set in bits<1:0>
            ;2592 ; rc[0e] - Contains base address of controller
            ;2593 ;
            ;2594 ; SIDE EFFECTS:
            ;2595 ;
            ;2596 ; d,va,sc are destroyed
            ;2597 ;--
            ;2598 SET_DIAG_MODE:
U 121D, 0010,0038,D9F8,0970,0284,721E  ;2599  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 121E, 0D00,003C,0180,0803,0000,121F  ;2600  va_va+4,d_dal.sc ; Shift specified mode into position
U 121F, 0000,003C,01E0,0800,0000,1220  ;2601  q_d    ; Save the diagnostic mode bits
U 1220, 0000,003C,0180,C800,0000,1221  ;2602  d[long]_cache.p  ; Read the contents of the CNFG register B
U 1221, 081D,0030,0180,0800,0000,1222  ;2603  d_d.or.q  ; Set the diagnostic mode bits
U 1222, 0000,003E,0180,A800,0000,0001  ;2604  cache.p_d[long],return1 ; Set the specified mode and return
            ;2605
```

```
            ;2606  .PAGE
            ;2607  .TOC "   SBI Unjam Routine"
            ;2608 ;**
            ;2609 ; FUNCTIONAL DESCRIPTION:
            ;2610 ;
            ;2611 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2612 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2613 ; and finally HOLD for the last 16 cycles.
            ;2614 ;
            ;2615 ; CALLING CONSTRAINT:
            ;2616 ;
            ;2617 ; =0
            ;2618 ;
            ;2619 ; SIDE EFFECTS:
            ;2620 ;
            ;2621 ; D Reg destroyed
            ;2622 ;--
            ;2623    ;
            ;2624    ; assert hold for 16 cycles
            ;2625    ;
            ;2626 CLRSBI:
            ;2627 SBI.UNJAM:
            ;2628  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 1223, 0818,0038,6580,8000,0010,105C  ;2629       clk.ubcc  ; set micro cc's for next cycle
            ;2630 =0
            ;2631 SBI.UNJAM.1:
            ;2632  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2633       clk.ubcc,z?, ; ...
U 105C, 0819,0100,0580,8000,0010,105C  ;2634       j/sbi.unjam.1 ; test for completion
            ;2635    ;
            ;2636    ; assert hold and unjam for 16 cycles
            ;2637    ;
            ;2638  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 105D, 0818,0038,6580,8800,0010,1064  ;2639       d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2640 =0
            ;2641 SBI.UNJAM.2:
            ;2642  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2643       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1064, 0819,0100,0580,8800,0010,1064  ;2644       z?,j/sbi.unjam.2 ; ...
            ;2645    ;
            ;2646    ; assert hold for 16 cycles
            ;2647    ;
            ;2648  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1065, 0818,0038,6580,8000,0010,106C  ;2649       clk.ubcc  ; and set cc's for next cycle
            ;2650 =0
            ;2651 SBI.UNJAM.3:
            ;2652  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2653       clk.ubcc,z?, ; ...
U 106C, 0819,0100,0580,8000,0010,106C  ;2654       j/sbi.unjam.3 ; ...
U 106D, 0000,003E,0180,0800,0000,0001  ;2655  return1   ; exit
```

```
            ;2656  .PAGE
            ;2657  .TOC "   RESET SUBTEST NUMBER"
            ;2658 ;**
            ;2659 ; FUNCTIONAL DESCRIPTION:
            ;2660 ;
            ;2661 ; Since some  of the tests will have to be restarted when two
            ;2662 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2663 ; the subtest counter to zero ( only for thoes tests that have
            ;2664 ; more than one subtest). This subroutine may also be necessary
            ;2665 ; when a test is being loop on.
            ;2666 ;
            ;2667 ; CALLING CONSTRAINT:
            ;2668 ;
            ;2669 ; =0
            ;2670 ; SIDE EFFECTS:
            ;2671 ;
            ;2672 ; D is destroyed
            ;2673 ;
            ;2674 ;--
            ;2675 RESET.SUBTST:
            ;2676  RC[0F]_0,   ; Reset subtest number
U 1224, 0003,003E,0180,09F8,0000,0001  ;2677   RETURN1   ; ...and return
```

```
            ;2678  .PAGE
            ;2679  .TOC "   Initialize the SBI"
            ;2680 ;++
            ;2681 ; FUNCTIONAL DESCRIPTION:
            ;2682 ;
            ;2683 ; This routine performs the following functions:
            ;2684 ;
            ;2685 ;  Executes an SBI Unjam
            ;2686 ;  Clears the SBI Fault Latch
            ;2687 ;  Clears the SBI Error Register
            ;2688 ;
            ;2689 ; CALLING CONSTRAINT:
            ;2690 ;
            ;2691 ; =0
            ;2692 ;
            ;2693 ; SIDE EFFECTS:
            ;2694 ;
            ;2695 ; D & Q Register destroyed
            ;2696 ;--
            ;2697 =0
            ;2698 SBI.INIT:
U 107C, 0000,003D,0180,0800,0000,1223  ;2699  call[sbi.unjam]  ; Unjam the SBI
U 107D, 0818,0038,C180,0800,0000,1225  ;2700  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1225, 0801,0028,6580,3C00,0000,1226  ;2701  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1226, 0F00,003C,6D80,3C00,0000,1227  ;2702  id[fault]_d,d_0
U 1227, 0000,003C,6D80,3C00,0000,1228  ;2703  id[fault]_d
U 1228, 0000,003E,6580,3C00,0000,0001  ;2704  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2705  .PAGE
            ;2706  .TOC "   Disable Cache Routine"
            ;2707 ;++
            ;2708 ; FUNCTIONAL DESCRIPTION:
            ;2709 ;
            ;2710 ; This routine disables cache hits by setting bits <16:15>
            ;2711 ; in the maintenance register.
            ;2712 ;
            ;2713 ; CALLING SEQUENCE:
            ;2714 ;
            ;2715 ; =0
            ;2716 ;
            ;2717 ; SIDE EFFECTS:
            ;2718 ;
            ;2719 ; D & Q Registers destroyed
            ;2720 ;--
            ;2721 DISABLE.CACHE:
U 1229, 0818.0038.4580.0800.0000.122A  ;2722  d_k[.8000]  ; ... and seed constant
U 122A, 0500.003C.0180.0800.0000.122B  ;2723  d_d.left  ; Generate final constant
U 122B, 0819.0030.4580.0800.0000.122C  ;2724  d_d.or.k[.8000]  ; ... = 00018000
U 122C, 0000.003E.7580.3C00.0000.0001  ;2725  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2726  .PAGE
            ;2727  .TOC "   Enable Cache"
            ;2728 ;++
            ;2729 ; FUNCTIONAL DESCRIPTION:
            ;2730 ;
            ;2731 ; This routine enables cache group 0 by clearing the force
            ;2732 ; miss bit in the maintenance register.
            ;2733 ;
            ;2734 ; CALLING CONSTRAINT:
            ;2735 ;
            ;2736 ; =0
            ;2737 ;
            ;2738 ; SIDE EFFECTS:
            ;2739 ;
            ;2740 ; D, Q AND SC Registers destroyed
            ;2741 ;--
            ;2742 ENABLE.CACHE:
U 122D, 0000.003C.75F0.2C00,0000.122E  ;2743  q_id[maint]  ; Get current maintenance register
U 122E, 0000.003C.6580.0800,0084.722F  ;2744  sc_k[.10]  ; Setup to clear bit 16
U 122F, 0801.2034.0180.0800,0000.1230  ;2745  d_q.and.mask  ; Clear bit 16
U 1230, 0000.003E.7580.3C00,0000.0001  ;2746  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2747  .PAGE
            ;2748  .TOC "   Wait Loop Routine"
            ;2749 ;++
            ;2750 ; FUNCTIONAL DESCRIPTION:
            ;2751 ;
            ;2752 ; As the name implies,  this subroutine is used to set up a wait
            ;2753 ; loop for a specified  number of cycles.  This may be necessary
            ;2754 ; when the micro-program has to wait for another event to finish.
            ;2755 ; When the subroutine is entered Register D should be holding the
            ;2756 ; desired numbered of cycles.
            ;2757 ;
            ;2758 ; CALLING CONSTRAINT:
            ;2759 ;
            ;2760 ; =0
            ;2761 ;
            ;2762 ; INPUTS:
            ;2763 ;
            ;2764 ; d - Contains number of cycles to wait
            ;2765 ; alu.z condition code must not be set
            ;2766 ;
            ;2767 ; SIDE EFFECTS:
            ;2768 ;
            ;2769 ; d is destroyed
            ;2770 ;--
            ;2771 WAIT.LOOP:
U 1231, 0018,0038,6180,0800,0010,1086  ;2772  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2773      ; ...is not set
            ;2774 =0
            ;2775 W.LOOP:
            ;2776  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1086, 0819,0100,0580,0800,0010,1086  ;2777       j/W.LOOP
U 1087, 0000,003E,0180,0800,0000,0001  ;2778  return1    ; exit
            ;2779
```

```
            ;2780  .PAGE
            ;2781  .TOC "   Check for Interrupt Routine"
            ;2782 ;++
            ;2783 ; FUNCTIONAL DESCRIPTION:
            ;2784 ;
            ;2785 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2786 ; If an interrupt is active, the following registers are setup:
            ;2787 ;  RC[8] - Gets the VECTOR register
            ;2788 ;  RC[7] - Gets the SBI ERROR register
            ;2789 ;  RC[6] - Gets the FAULT register
            ;2790 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2791 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2792 ;
            ;2793 ; CALLING CONSTRAINT:
            ;2794 ;
            ;2795 ; =00
            ;2796 ;
            ;2797 ; SIDE EFFECTS:
            ;2798 ;
            ;2799 ; D & Q are destroyed
            ;2800 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2801 ;--
            ;2802 CHECK_INTERRUPT:
            ;2803    ;
            ;2804    ; First, enable the fault and RDS/CRD interrupts
            ;2805    ;
U 1232, 0818,0038,4580,0800,0000,1233  ;2806  d_k[.8000]  ; Get data to set interrupt enable
            ;2807  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1233, 0500,003C,6580,3C00,0000,1234  ;2808       d_d.left
U 1234, C100,003C,0180,0800,0000,1235  ;2809  d_d.left2  ; D = 40000
U 1235, 0000,003C,6D80,3C00,0000,1236  ;2810  id[fault]_d  ; Set fault interrupt enable
            ;2811  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1236, 0F03,003C,0180,09A8,4000,1237  ;2812       rc[5]_0  ; and clear interrupt occurred flag
U 1237, 0000,003C,3D80,3C00,0000,1238  ;2813  id[psl]_d  ; Clear the PSL
U 1238, 0000,003C,6580,3C00,0000,1239  ;2814  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1239, 0000,003C,6D80,3C00,0000,123A  ;2815  id[fault]_d  ; Clear FAULT Interrupt Enable
U 123A, 0000,0E3C,0180,0800,0000,1004  ;2816  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2817 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2818  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2819  return1   ; No interrupt
            ;2820 =111
            ;2821 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,123B  ;2822  q_id[vector]  ; Get ID Vector Register
U 123B, 0001,203C,65F0,2DC0,0000,123C  ;2823  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 123C, 0001,203C,6DF0,2DB8,0000,123D  ;2824  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 123D, 0001,203C,0180,09B0,0000,123E  ;2825  rc[6]_q   ; Save falut reg
U 123E, 0018,003A,0580,09A8,0000,0002  ;2826  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2827  .PAGE
            ;2828  .TOC "   CHECK UTRAP"
            ;2829 ;++
            ;2830 ;FUNCTIONAL DESCRIPTION:
            ;2831 ;
            ;2832 ; This subroutine is used to check wether a Utrap occured.
            ;2833 ; It takes the contents of the Save Utrap flags register
            ;2834 ; R[0E], and compares them to the contrnts of the Utrap
            ;2835 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2836 ; according to the results of the test  (i.e., contents of
            ;2837 ; these registers are equal or not).
            ;2838 ; CALLING CONSTRAINT:
            ;2839 ;
            ;2840 ; =00
            ;2841 ;
            ;2842 ; INPUTS:
            ;2843 ;
            ;2844 ; R[0E]- Saved Utrap Mask
            ;2845 ; R[0F]- Expected Utrap Mask
            ;2846 ;
            ;2847 ;--
            ;2848 CHECK_UTRAP:
U 123F, 0800.003C.0180.0A70.0000.1240  ;2849  D_R[0E]    ; Reg D is loaded with the trap mask
U 1240, 0001.003C.0180.09C0.0000.1241  ;2850  RC[08]_D   ; Save expected Utrap flag for error logout
U 1241, 0800.003C.0180.0A78.0000.1242  ;2851  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1242, 0001.003C.0180.09B8.0000.1243  ;2852  RC[07]_D   ; Save in RC[07]
            ;2853  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1243, 000D.0020.0180.0A70.0010.1244  ;2854          CLK.UBCC
U 1244, 0003.013C.0180.0AF8.0000.10F0  ;2855  Z?,R[0F]_0  ; Branch if no traps
U 10F0, 0000.003E.0180.0800.0000.0002  ;2856 =0 RETURN2   ; Utrap occured
U 10F1, 0000.003E.0180.0800.0000.0001  ;2857  RETURN1    ; No Utrap  return
            ;2858 ;
```

```
            ;2859  .PAGE
            ;2860  .TOC "   CHECK SBI.ERR"
            ;2861 ;**
            ;2862 ; FUNCTIONAL DESCRIPTION:
            ;2863 ;
            ;2864 ; This subroutine can be used to see whether the SBI.ERR Register
            ;2865 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;2866 ; routine will have to pass to this subroutine in the SC register
            ;2867 ; the number of the error bit to checked.
            ;2868 ;
            ;2869 ; CALLING CONSTRAINT:
            ;2870 ;
            ;2871 ; =00
            ;2872 ;
            ;2873 ; INPUTS:
            ;2874 ;
            ;2875 ; SC with the number of the error bit to be checked
            ;2876 ;
            ;2877 ; OUTPUTS:
            ;2878 ;
            ;2879 ; RC[08]-Contains the SBI.ERROR Register
            ;2880 ;--
            ;2881 CHECK_SBI_ERR:
U 1245, 0000.003C.65F0.2C00.0000.1246  ;2882  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 1246, 0001.203C.0180.09C0.0000.1247  ;2883  RC[08]_Q  ; Save for error logout
            ;2884  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 1247, 0001.2024.0180.0800.0010.1248  ;2885   CLK.UBCC ; ...
U 1248, 0000.013C.0180.0800.0000.10F2  ;2886  Z?   ; Check condition codes
U 10F2, 0000.003E.0180.0800.0000.0002  ;2887 =0 RETURN2   ; Bit is set
U 10F3, 0000.003E.0180.0800.0000.0001  ;2888  RETURN1   ; Bit is not set
```

```
            ;2889  .PAGE
            ;2890  .TOC "   Set Trap Mask"
            ;2891 ;++
            ;2892 ; FUNCTIONAL DESCRIPTION:
            ;2893 ;
            ;2894 ; This routine is used to set a bit in the Micro Trap Mask
            ;2895 ; R[0F].
            ;2896 ;
            ;2897 ; CALLING CONSTRAINT:
            ;2898 ;
            ;2899 ; =0
            ;2900 ;
            ;2901 ; INPUTS:
            ;2902 ;
            ;2903 ; SC - Contains bit number to set.
            ;2904 ;
            ;2905 ; OUTPUTS:
            ;2906 ;
            ;2907 ; rc[0E] - Contains the current trap mask
            ;2908 ;
            ;2909 ; SIDE EFFECTS:
            ;2910 ;
            ;2911 ; d regsiter is destroyed
            ;2912 ;--
            ;2913 SET_TRAP_MASK:
U 1249, 0800.003C.0180.0A78.0000.124A  ;2914  d_r[0f]
U 124A, 0801.001C.0180.0AF8.0000.124B  ;2915  r[0f]_d.ornot.mask,d_alu ; Set mask
U 124B, 0001.003E.0180.0AF0.0000.0001  ;2916  r[0E]_d,return1    ; Save mask and return
```

```
            ;2917  .PAGE
            ;2918  .TOC "   FIND MS780-E MEMORY"
            ;2919 ;++
            ;2920 ; FUNCTIONAL DESCRIPTION:
            ;2921 ;
            ;2922 ;          The diagnostic is designed to handle up to 4 (four)
            ;2923 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2924 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2925 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2926 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2927 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2928 ;     first MS780-E found will have its starting address set to 0
            ;2929 ;     (zero).
            ;2930 ;     All the other MS780-E/H's found will have their starting address
            ;2931 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2932 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2933 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2934 ;     current MS780-E unit under test.
            ;2935 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2936 ;     set to maximum.
            ;2937 ;
            ;2938 ; CALLING CONSTRAINT:
            ;2939 ;
            ;2940 ; =00
            ;2941 ;
            ;2942 ; OUTPUTS:
            ;2943 ;
            ;2944 ; rc[0e] - Contains address of Configuration Register
            ;2945 ; r[0]  - Contains TR level
            ;2946 ;
            ;2947 ; SIDE EFFECTS:
            ;2948 ;
            ;2949 ; D register is destroyed.
            ;2950 ;--
            ;2951 =0
            ;2952 FIND.MS780E:
U 10F4, 0000,003D,0180,0800,0000,107C  ;2953  call[sbi.init]  ; Make sure SBI is enabled
U 10F5, 0003,003C,0180,0988,0000,124C  ;2954  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 124C, 0818,0038,1980,0800,0000,124D  ;2955  d_k[zero]  ; Clear MS780-E/H counter
U 124D, 0000,003C,F980,3C00,0000,124E  ;2956  id[3e]_d  ; ...
U 124E, 0018,0038,0580,0A80,0000,124F  ;2957  r[0]_k[.1]  ; Init TR counter
U 124F, 0F03,003C,0180,09F0,0000,10F6  ;2958  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2959      ; ...CNFG A Reg
            ;2960 =0
            ;2961 FIND.MEM.LOOP:
U 10F6, 0800,003D,0180,0A00,0000,1275  ;2962  d_r[0],call[generate.io]; Generate address of TR level
U 10F7, 0001,003C,0180,09F0,0200,10F8  ;2963  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10F8, 0000,003D,8980,0800,0084,7249  ;2964 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2965  d[long]_cache.p, ; Read the specified tr level
U 10F9, 0000,003C,0180,C970,0000,1250  ;2966   lc_rc[0e]  ; ...
U 1250, 0000,003C,0180,0A78,0010,1251  ;2967  alu_r[0f],clk.ubcc ; Check for no response
U 1251, 0001,013C,0180,0880,0082,10FA  ;2968  z?,sc_d,la_re[0] ; Branch if no adapter here
U 10FA, 0000,003C,0180,0800,0000,1252  ;2969 =0 j/check.adpt.code ; Check for a memory
U 10FB, 0000,003C,0180,0800,0000,1271  ;2970  j/find.mem.lp1  ; Try next tr
            ;2971 CHECK.ADPT.CODE:
```

```
U 1252, 0000,003C,1D80,0800,1404,7253  ;2972  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1253, 0000,163C,0180,0800,0000,1028  ;2973  state7-4?  ; Branch on the adapter type
U 1028, 0000,003C,8980,0800,0084,7254  ;2974 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1029, 0000,003C,8980,0800,0084,7255  ;2975  j/ms780.c,sc_k[.d] ; MS780-C
U 102A, 0000,003C,0180,0800,0000,1271  ;2976  j/find.mem.lp1  ; Not a memory
U 102B, 0000,003C,0180,0800,0000,1271  ;2977  j/find.mem.lp1  ; Not a memory
U 102C, 0000,003C,6580,0800,0084,725A  ;2978  j/ma780.max,sc_k[.10] ; MA780
U 102D, 0000,003C,0180,0800,0000,1271  ;2979  j/find.mem.lp1  ; Not a memory
U 102E, 0010,0038,0180,09F0,0000,125E  ;2980  rc[0e]_lc,j/found.ms780e ; MS780-E
U 102F, 0010,0038,0180,09F0,0000,125E  ;2981  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2982   ;
          ;2983   ; Found an MS780-A or -C. Set the starting address
          ;2984   ; to maximum.
          ;2985   ;
U 1254, 0818,0038,5D80,0800,0000,1256  ;2986 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1255, 0818,0038,0580,0800,0000,1256  ;2987 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
          ;2988 MS780.COMMON:
U 1256, 0819,0030,7180,0800,0000,1257  ;2989  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1257, 0000,003C,01F8,0803,0080,D258  ;2990  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1258, 0D00,003C,0180,0800,0000,1259  ;2991  d_dal.sc  ; Put data in bits<29:14>
          ;2992  cache.p_d[long], ; Set the starting address to maximum
U 1259, 0000,003C,0180,A800,0000,1271  ;2993       j/find.mem.lp1 ; and continue TR scan
          ;2994   ;
          ;2995   ; Found an MA780. Set the starting address to maximum
          ;2996   ;
          ;2997 MA780.MAX:
U 125A, 0818,0038,39F8,0803,0000,125B  ;2998  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 125B, 0D00,003C,0180,0803,0000,125C  ;2999  d_dal.sc,va_va+4 ; Put in proper position
U 125C, 0000,003C,0180,0803,0000,125D  ;3000  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;3001  cache.p_d[long], ; Set starting address to maximum
U 125D, 0000,003C,0180,A800,0000,1271  ;3002       j/find.mem.lp1
          ;3003 ;
          ;3004 ; Found an MS780E
          ;3005 ;
          ;3006 FOUND.MS780E:
U 125E, 0000,003C,F9F0,2C00,0000,125F  ;3007  q_id[3e]  ; Set up to see how many MS780-E's
          ;3008  alu_q.xor.k[.3], ; Are there Maximum units?
U 125F, 0019,2020,0D80,0800,0010,1260  ;3009   clk.ubcc ; Check condition codes
U 1260, 0000,013C,0180,0800,0000,10FC  ;3010  z?   ; Are we at maximum units for system
U 10FC, 0000,003C,0180,0800,0000,1261  ;3011 =0 J/ms780e.tst  ; No go find how many units ther are
U 10FD, 0818,0038,C180,0800,0000,10FE  ;3012  d_k[.ffff]  ; Yes set address to maximum
U 10FE, 0000,003D,0180,0800,0000,1279  ;3013 =0 call[set.start.addr] ; .....
U 10FF, 0000,003C,0180,0800,0000,1271  ;3014  j/find.mem.lp1  ; Go check to see if we are done
          ;3015 ;
          ;3016 MS780E.TST:
          ;3017  alu_rc[01],  ; Check "Found MS780E Flag"
U 1261, 0010,0038,0180,0908,0010,1262  ;3018   clk.ubcc ; Check condition codes
U 1262, 0810,0138,0180,0970,0000,110A  ;3019  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;3020 =0 j/not.first.ms780e, ; No
U 110A, 0818,0038,C180,0800,0000,1112  ;3021   d_k[.ffff] ; Set starting address
U 110B, 0001,003C,0180,0988,0000,1263  ;3022  rc[01]_d  ; Yes put base address in rc[01]
U 1263, 0818,0038,7980,0800,0000,1264  ;3023  d_k[.30]  ; Set up SC to point to first unit
U 1264, 0019,0014,F580,0800,0082,1265  ;3024  alu_d+k[.a],sc_alu ; ...
U 1265, 0810,0038,0180,0908,0000,1266  ;3025  d_rc[01]  ; Put base address of unit in D
U 1266, 0000,003C,0180,3400,0000,1267  ;3026  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1267, 0F00,003C,0180,0800,0000,1110  ;3027  d_0   ; Prepare to set starting address to Zero
U 1110, 0000,003D,0180,0800,0000,1279  ;3028 =0 call[set.start.addr] ; Go do it
        ;3029  j/find.mem.lp1,  ; Check to see if we are done
U 1111, 0003,003C,0180,09E8,0000,1271  ;3030   rc[0d]_0 ; ....
        ;3031 =0
        ;3032 NOT.FIRST.MS780E:
U 1112, 0000,003D,0180,0800,0000,1279  ;3033  call[set.start.addr] ; Go set starting address to maximum
U 1113, 0000,003C,F9F0,2C00,0000,1268  ;3034  q_id[3e]  ; Get the number of MS780-Es found
U 1268, 0819,2014,0580,0800,0000,1269  ;3035  d_q+k[.1]  ; Increment this counter
U 1269, 0000,003C,F980,3C00,0000,126A  ;3036  id[3e]_d  ; Save the counter
U 126A, 0018,0038,11C0,0800,0000,126B  ;3037  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;3038      ; ...were the I/O base addresses will be stored
U 126B, 081D,2000,7980,0800,0000,126C  ;3039  d_q-d,k[.30]  ; ... adjust pointer
U 126C, 0819,0014,7980,0800,0000,126D  ;3040  d_d+k[.30]  ; Point the SC to the ID register
U 126D, 0000,003C,F580,0800,0000,126E  ;3041  k[.a]   ; ...to receive I/O base address
U 126E, 0019,0014,F580,0800,0082,126F  ;3042  alu_d+k[.a],sc_alu ; ...
U 126F, 0810,0038,0180,0970,0000,1270  ;3043  d_rc[0e]  ; Get base address to d
U 1270, 0000,003C,0180,3400,0000,1271  ;3044  id(sc)_d  ; Move base address to ID pointed to by SC
        ;3045   ;
        ;3046   ; Try next tr
        ;3047   ;
        ;3048 FIND.MEM.LP1:
U 1271, 0818,0014,0580,0A80,0000,1272  ;3049  r[0]_la+k[.1],d_alu ; Increment TR level
        ;3050  alu_d.and.k[.f],
U 1272, 0019,0034,6180,0800,0010,1273  ;3051      clk.ubcc  ; Check if all done
U 1273, 0000,013C,0180,0800,0000,1114  ;3052  z?   ; Branch if yes
U 1114, 0000,003C,0180,0800,0000,10F6  ;3053 =0 j/find.mem.loop  ; Try next TR
U 1115, 0810,0038,0180,0908,0010,1274  ;3054  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1274, 0000,013C,0180,0800,0000,1116  ;3055  z?   ; Check condition codes
U 1116, 0001,003E,0180,09F0,0000,0002  ;3056 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1117, 0000,003E,0180,0800,0000,0001  ;3057  return1   ; No MS780E found
        ;3058
```

```
            ;3059  .PAGE
            ;3060  .TOC "   Generate IO Address"
            ;3061 ;++
            ;3062 ; FUNCTIONAL DESCRIPTION:
            ;3063 ;
            ;3064 ; This routine is used to generate an SBI Configuration Register
            ;3065 ; address from a TR level.
            ;3066 ;
            ;3067 ; CALLING CONSTRAINT:
            ;3068 ;
            ;3069 ; =0
            ;3070 ;
            ;3071 ; INPUTS:
            ;3072 ;
            ;3073 ; D - Contains TR level
            ;3074 ;
            ;3075 ; OUTPUTS:
            ;3076 ;
            ;3077 ; D - Contains SBI IO address
            ;3078 ;--
            ;3079 GENERATE.IO:
U 1275, 0000,003C,89F8,0800,0084,7276  ;3080  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1276, 0D00,003C,8D80,0800,0084,7277  ;3081  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 1277, 0000,003C,0980,0800,0084,B278  ;3082  sc_sc-k[.2]  ; insert bit 29
U 1278, 0801,001E,0180,0800,0000,0001  ;3083  d_d.ornot.mask,return1 ; and return
```

```
            ;3084  .PAGE
            ;3085  .TOC "   Set Starting Address"
            ;3086 ;++
            ;3087 ; FUNCTIONAL DESCRIPTION:
            ;3088 ;
            ;3089 ; This routine is used to set the starting address of the
            ;3090 ; memory to the specified value.
            ;3091 ;
            ;3092 ; CALLING CONSTRAINT:
            ;3093 ;
            ;3094 ; =0
            ;3095 ;
            ;3096 ; INPUTS:
            ;3097 ;
            ;3098 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3099 ;     (B Register Image divided by 2**16)
            ;3100 ; rc[0e] - Contains Address of Register A
            ;3101 ;
            ;3102 ; OUTPUTS:
            ;3103 ;
            ;3104 ; d - Contains aligned starting address (B Register Image)
            ;3105 ;
            ;3106 ; SIDE EFFECTS:
            ;3107 ;
            ;3108 ; d,q,va, and sc are destroyed
            ;3109 ;--
            ;3110 SET.START.ADDR:
U 1279, 0010,0038,6580,0970,0284,727A  ;3111  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 127A, 0000,003C,01F8,0803,0000,127B  ;3112  va_va+4,q_0  ; Set address to Register B
            ;3113  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 127B, 0D00,003C,0980,0800,0084,B27C  ;3114      sc_sc-k[.2]  ; Setup to insert bit 14
U 127C, 0801,001C,0180,0800,0000,127D  ;3115  d_d.ornot.mask  ; Insert Write Enable bit
U 127D, 0000,003E,0180,A800,0000,0001  ;3116  cache.p_d[long],return1 ; Set the starting address and return
```

```
                ;3117  .PAGE
                ;3118  .TOC "   ENABLE NEXT MS780-E"
                ;3119 ;++
                ;3120 ; FUNCTIONAL DESCRIPTION:
                ;3121 ;
                ;3122 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
                ;3123 ;    They will be tested  separately, according to the TR level at which
                ;3124 ;    they are located,  starting with the one  at the lowest level first.
                ;3125 ;          When  a test has finished  with the first unit, it will have to call
                ;3126 ;    this specific  routine  to see  if any other MS780-E unit is present
                ;3127 ;    on the SBI. If more units are  present, the first one  will be  dis-
                ;3128 ;    abled  by  setting  its  starting  address to maximum, the next unit
                ;3129 ;    will be  enabled  by  setting its starting address to 0 and the test
                ;3130 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
                ;3131 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
                ;3132 ;    ID registers 3A through 3D and  a count of the Total number of units
                ;3133 ;    found - 1 in register 3E. In this subroutine the SC register is used
                ;3134 ;    as  a  pointer  to  ID  registers 3A through 3D.
                ;3135 ;
                ;3136 ; CALLING CONSTRAINT:
                ;3137 ;
                ;3138 ;  =00
                ;3139 ;
                ;3140 ;--
                ;3141 ENABLE.NEXT.MS780E:
U 127E, 0000.003C.F9F0.2C00.0000.127F  ;3142  Q_ID[3E]  ; Get total number of MS780E to Q
                ;3143  ALU_Q,    ; Check to see if it is zero
U 127F, 0001.203C.0180.0800.0010.1280  ;3144      CLK.UBCC  ; Check Condition Codes
U 1280, 0000.013C.0180.0800.0000.1118  ;3145  Z?    ; ...for zero
U 1118, 0000.003C.0180.0800.0000.1281  ;3146 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1119, 0000.003E.0180.0800.0000.0002  ;3147  RETURN2    ; Zero No more units to test
                ;3148 ENABLE.NEXT:
U 1281, 0818.0038.C180.0800.0000.111A  ;3149  D_K[.FFFF]  ; Load D with Maximum Starting address
U 111A, 0000.003D.0180.0800.0000.1279  ;3150 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 111B, 0818.0038.7980.0800.0000.1282  ;3151  D_K[.30]  ; Prepare to point to the ID register holding
                ;3152      ; ...the I/O Base address of the next MS780-E
U 1282, 0819.0014.1180.0800.0000.1283  ;3153  D_D+K[.4]  ; ...
U 1283, 0000.003C.F580.0800.0000.1284  ;3154  K[.A]    ; ...
U 1284, 0819.0014.F580.0800.0000.1285  ;3155  D_D+K[.A]  ; ...
U 1285, 0000.003C.F9F0.2C00.0000.1286  ;3156  Q_ID[3E]  ; Get MS780-E Counter
                ;3157  D_D-Q,    ; D now holds the pointer to the ID register
U 1286, 081D.0000.0180.0800.0082.1287  ;3158    SC_ALU  ; ...
U 1287, 0000.003C.01F0.2400.0000.1288  ;3159  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1288, 0001.203C.0180.09F0.0000.1289  ;3160  RC[0E]_Q  ; Save it also in RC[0E]
U 1289, 0F03.003C.0180.09E8.0000.111C  ;3161  RC[0D]_0,D_0  ; Set starting address to zero
U 111C, 0000.003D.0180.0800.0000.1279  ;3162 =0 CALL[SET.START.ADDR] ; ....
U 111D, 0F00.003C.F9F0.2C00.0000.128A  ;3163  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 128A, 081D.2008.0180.0800.0000.128B  ;3164  D_Q-D-1    ; Subtract 1 from total
U 128B, 0000.003C.F980.3C00.0000.111E  ;3165  ID[3E]_D  ; Adjust the pointer
U 111E, 0000.003D.0180.0800.0000.1224  ;3166 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 111F, 0000.003E.0180.0800.0000.0001  ;3167  RETURN1    ; Tell caller another unit was found
                ;3168
```

```
;3169  .PAGE
;3170  .TOC "   FIND CONTROLLERS"
;3171 ;
;3172 ;////////////////////////////////////////////////////////////////////////////////
;3173 ;++
;3174 ;
;3175 ; Functional Description:
;3176 ; ----------------------
;3177 ;
;3178 ;
;3179 ;      This subroutine is called by tests that  need  to  know  how
;3180 ; many  controllers  are available on the MS780-E/H currently under
;3181 ; test.  It makes  use  of  subroutine  SELECT.CNTRLR  to  put  the
;3182 ; controllers in different Interleave Modes.
;3183 ; The subroutine returns control to the calling routine  by  either
;3184 ; one  of:   Return1, Return2, Return3 or Return4.  It is up to the
;3185 ; test to decode this information as necessary.
;3186 ; Since some Tests require this feature, the subroutine also  finds
;3187 ; the  memory  size  available  on each controller and leaves it in
;3188 ; registers R[0] and R[1].
;3189 ;
;3190 ;
;3191 ; Calling Constraint: =000
;3192 ; ------------------
;3193 ;
;3194 ;
;3195 ; Inputs:
;3196 ; ------
;3197 ;
;3198 ; RC[0E] MUST HOLD THE I/O BASES ADDRESS OF THE MS780-E/H
;3199 ; UNDER TEST
;3200 ;
;3201 ; Outputs:
;3202 ; -------
;3203 ;
;3204 ; - Return1, Return2, Return3 or Return4 depending on
;3205 ;   how many Controllers were found
;3206 ;
;3207 ; - R[0] Size of Memory on the Lower Controller
;3208 ;   R[1] Size of Memory on the Upper Controller
;3209 ;
;3210 ; Side Effects:
;3211 ; ------------
;3212 ;
;3213 ; The contents of the following registers will be lost:
;3214 ;
;3215 ; D, Q, SC, R[0], R[1]
;3216 ;
;3217 ;--
;3218 ;////////////////////////////////////////////////////////////////////////////////
;3219 ;
;3220 =00
;3221 FIND.CNTRLRS:
;3222  D_K[ZERO],   ; Attempt to select the Lower
U 1040, 0818,0039,1980,0800,0000,1207  ;3223   CALL[SELECT.CNTRLR] ; ...Controller
```

```
U 1041, 0000,003C,0180,0800,0000,1050  ;3224  J/LOW.MIS.X    ; Lower Controller Misconfigured
U 1042, 0000,003D,0180,0800,0000,128C  ;3225  CALL[GET.MEMORY.SIZE]  ; Lower Controller has been  configured
        ;3226        ; ...find its memory size
U 1043, 0001,003C,0180,0A80,0000,1048  ;3227  R[0]_D     ; Save Memory size in R[0]
        ;3228 =00 D_K[.2],    ; Attempt to select the Upper
U 1048, 0818,0039,0980,0800,0000,1207  ;3229   CALL[SELECT.CNTRLR] ; ...Controller
U 1049, 0000,003E,0180,0800,0000,0002  ;3230  RETURN2     ; Upper Controller Misconfigured.
        ;3231        ; ...Let caller know that ONLY the
        ;3232        ; ...Lower Controller was found
U 104A, 0000,003D,0180,0800,0000,128C  ;3233  CALL[GET.MEMORY.SIZE]  ; Upper Controller was Configured so
        ;3234        ; ...find its memory size
        ;3235  R[1]_D,     ; Save Memory Size in R[1]
U 104B, 0001,003E,0180,0A88,0000,0004  ;3236   RETURN4    ; Let Caller know that BOTH Controllers
        ;3237        ; ...were found
        ;3238 =00
        ;3239 LOW.MIS.X:
        ;3240  D_K[.2],    ; Lower Controller was not found,
U 1050, 0818,0039,0980,0800,0000,1207  ;3241   CALL[SELECT.CNTRLR] ; ...so attempt to Configure the
        ;3242        ; ...Upper Controller
U 1051, 0000,003E,0180,0800,0000,0001  ;3243  RETURN1     ; Tell Caller that NONE of the
        ;3244        ; ...Controllers were found
U 1052, 0000,003D,0180,0800,0000,128C  ;3245  CALL[GET.MEMORY.SIZE]  ; Upper Controller was found,
        ;3246        ; ...go get its memory size
        ;3247  R[1]_D,     ; Save the memory size in R[1]
U 1053, 0001,003E,0180,0A88,0000,0003  ;3248   RETURN3    ; Let Caller know that ONLY the
        ;3249        ; ...Upper Controller was found
        ;3250
```

```
            ;3251  .PAGE
            ;3252  .TOC "  GET MEMORY SIZE"
            ;3253 ;*****************************************************************
            ;3254 ;++
            ;3255 ;
            ;3256 ; Functional Description:
            ;3257 ; ----------------------
            ;3258 ;
            ;3259 ; This subroutine can be used to get the size of the memory.
            ;3260 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3261 ; increments.
            ;3262 ;
            ;3263 ; Calling Constraint: =0
            ;3264 ; ------------------
            ;3265 ;
            ;3266 ;
            ;3267 ; Inputs:
            ;3268 ; ------
            ;3269 ;
            ;3270 ; RC[0E] Must have the Address of CNFG Register A
            ;3271 ;
            ;3272 ; Outputs:
            ;3273 ; -------
            ;3274 ;
            ;3275 ; Register D will contain upon exit the size of the
            ;3276 ; memory aligned on Mega-byte boundary.
            ;3277 ;
            ;3278 ; Side Effects:
            ;3279 ; ------------
            ;3280 ;
            ;3281 ; The contents of the following registers will be lost:
            ;3282 ;
            ;3283 ; D, SC, Q
            ;3284 ;
            ;3285 ;--
            ;3286 ;*****************************************************************
            ;3287 GET.MEMORY.SIZE:
U 128C, 0010,0038,0180,0970,0200,128D  ;3288  VA_RC[0E]    ; Point to CNFG Reg A
U 128D, 0000,003C,0180,C800,0000,128E  ;3289  D[LONG]_CACHE.P    ; Read the register
U 128E, 001B,0000,D980,0800,0082,128F  ;3290  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 128F, 0D00,003C,0180,0800,0000,1290  ;3291  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 1290, 0819,0034,5580,0800,0000,1060  ;3292  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3293 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1060, 0000,003D,01E0,0800,0000,1211  ;3294   Q_D    ; Save Memory size bits in Q
U 1061, 0000,003D,0D80,0800,0084,703C  ;3295  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3296       ; ...arrays on the controller
            ;3297 ;
            ;3298 ; The duplication of the next two return status is necessary so the returning
            ;3299 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3300 ; modifications.
            ;3301 ;
U 1062, 0819,2014,0580,0800,0000,1063  ;3302  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3303       ; ...(found 1 Meg arrays)
U 1063, 0819,2014,0580,0800,0000,1291  ;3304  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3305       ; ...(found 4 Meg arrays)
```

```
U 1291, 0000,003C,21F8,0800,0084,7292  ;3306  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
        ;3307  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 1292, 0D00,003E,0180,0800,0000,0001  ;3308   RETURN1   ; Return with memory size in Register D
        ;3309
```

```
;3310  .PAGE
;3311  .TOC "    MS780-E/H CNFG REG CLEANUP"
;3312 ;*************************************************************
;3313 ;**
;3314 ;
;3315 ; Functional Description:
;3316 ; ----------------------
;3317 ;
;3318 ;       This subroutine is used to clear all error conditions
;3319 ; in the MS780-E/H Configuration/Status/Error Registers.
;3320 ; Bits beeing cleared are:
;3321 ;
;3322 ; CNFG Regsiter B <11:00>
;3323 ;
;3324 ; CNFG Register C and D <31:28> and <10:06>
;3325 ;   ---
;3326 ;
;3327 ;
;3328 ; Calling Constraint: =0
;3329 ; ------------------
;3330 ;
;3331 ;
;3332 ; Inputs:
;3333 ; ------
;3334 ;
;3335 ; RC[0E] MUST contain the I/O base address of
;3336 ; the MS780-E/H currently under test
;3337 ;
;3338 ; Outputs:
;3339 ; -------
;3340 ;
;3341 ; NO specific outputs.
;3342 ; Above mentioned bits will be cleared.
;3343 ;
;3344 ; Side Effects:
;3345 ; ------------
;3346 ;
;3347 ; The contents of the following registers will be lost
;3348 ; Q, D, SC, VA
;3349 ;
;3350 ;
;3351 ;--
;3352 ;*************************************************************
;3353 MS780E.CLEANUP:
U 1293, 0010,0038,0180,0970,0200,1294  ;3354  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
        ;3355  D_0,      ; Get data to clear Read/Write bits
U 1294, 0F00,003C,0180,0803,0000,1295  ;3356   VA_VA+4    ; Point to CNFG Reg B
U 1295, 0000,003C,0180,A800,0000,1296  ;3357  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 1296, 0818,0038,7980,0800,0000,1297  ;3358  D_K[.30]    ; Prepare to form x30000780 to clear
        ;3359       ; ...CNFG Reg's C and D
U 1297, 0B00,003C,0180,0800,0000,1298  ;3360  D_D.SWAP    ; D = x30000000
        ;3361  Q_D,      ; Save in the Q Reg
U 1298, 0818,0038,CDE0,0800,0000,1299  ;3362   D_K[.F0]   ; D = xF0
U 1299, 0000,003C,0D80,0800,0084,729A  ;3363  SC_K[.3]    ; Must shift D Reg left 3 bits to get
        ;3364       ; ...x780
```

```
U 129A, 0D00,003C,0180,0800,0000,129B  ;3365  D_DAL.SC    ; D = x780
        ;3366  D_D.OR.Q,    ; D = x30000780
U 129B, 081D,0030,0180,0803,0000,129C  ;3367   VA_VA+4    ; Point to CNFG Reg C
U 129C, 0000,003C,0180,A800,0000,129D  ;3368  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 129D, 0000,003C,0180,0800,0000,129E  ;3369  NOP
U 129E, 0000,003C,0180,A800,0000,129F  ;3370  CACHE.P_D[LONG]   ; Do second write to make sure that
        ;3371       ; ...uSequencer Parity error bit
        ;3372       ; ...is clear
U 129F, 0000,003C,0180,0803,0000,12A0  ;3373  VA_VA+4    ; Point to CNFG Reg D
U 12A0, 0000,003C,0180,A800,0000,12A1  ;3374  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 12A1, 0000,003C,0180,0800,0000,12A2  ;3375  NOP
        ;3376  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
        ;3377       ; ...Parity error bit
U 12A2, 0000,003E,0180,A800,0000,0001  ;3378   RETURN1    ; Return to caller
        ;3379
        ;3380
```

```
            ;3381   .PAGE
            ;3382   .TOC "    INIT ROUTINE"
            ;3383   ;=====================================================================
            ;3384   ;
            ;3385   ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;3386   ;    PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;3387   ;     FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;3388   ;
            ;3389   ;=====================================================================
            ;3390   ;
U 12A3, 0003.003C.0180.0AF8.0000.1120   ;3391 INIT: R[0F]_0     ;CLEAR U-TRAPS
U 1120, 0000.003D.0180.0800.0000.1223   ;3392 =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 1121, 0000.003C.0180.0800.0000.1122   ;3393  NOP
U 1122, 0000.003D.0180.0800.0000.12A6   ;3394 =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;3395       ;  INTRUPT REG
U 1123, 0000.003C.0180.0800.0000.1124   ;3396  NOP
U 1124, 0000.003D.0180.0800.0000.12AE   ;3397 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 1125, 0000.003C.0180.0800.0000.1126   ;3398  NOP
U 1126, 0000.003D.0180.0800.0000.12B2   ;3399 =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 1127, 0000.003C.0180.0800.0000.1128   ;3400  NOP
U 1128, 0000.003D.0180.0800.0000.12A9   ;3401 =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 1129, 0000.003C.0180.0800.0000.112A   ;3402  NOP
U 112A, 0000.003D.0180.0800.0000.12B6   ;3403 =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 112B, 0000.003C.0180.0800.0000.112C   ;3404  NOP
U 112C, 0000.003D.0180.0800.0000.12BA   ;3405 =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 112D, 0818.0038.4580.0800.0000.112E   ;3406  D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;3407 =0 CALL[DISABLE.CACHE],   ;SHUT CACHE OFF
U 112E, 0000.003D.7980.3C00.0000.1229   ;3408    ID[PARITY]_D
U 112F, 0F00.003C.0180.0800.0000.12A4   ;3409  D_0
U 12A4, 0000.003C.4980.3C00.0000.12A5   ;3410  ID[TBER0]_D    ;SHUT TB OFF
U 12A5, 0000.003E.7180.3C00.0000.0001   ;3411  ID[COMP]_D,RETURN1   ;DISABLE SILO INTERRUPTS
            ;3412
```

```
            ;3413  .PAGE
            ;3414  .TOC "    SETPSL ROUTINE"
            ;3415 SETPSL: ;=================================================================
            ;3416  ;
            ;3417  ;DROP THE CPU IPR LEVEL TO
            ;3418  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;3419  ; PENDING
            ;3420  ;
            ;3421  ;=================================================================
            ;3422  ;
U 12A6, 0F00.003C.0180.0800.0000.12A7  ;3423  D_0
U 12A7, 0000.003C.3980.3C00.0000.12A8  ;3424  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 12A8, 0000.003E.3D80.3C00.0000.0001  ;3425  RETURN1,ID[PSL]_D
            ;3426
```

```
            ;3427  .PAGE
            ;3428  .TOC "   CLRTIM ROUTINE"
            ;3429 CLRTIM: ;=================================================================
            ;3430  ;
            ;3431  ;CLEAR THE CP TIMEOUT BIT IN THE
            ;3432  ; SBI ERROR REGISTER
            ;3433  ;
            ;3434  ;=================================================================
            ;3435  ;
U 12A9, 0818.0038.0580.0800.0000.12AB  ;3436  D_K[.1]
U 12AB, 0000.003C.85F8.0800.0084.72AC  ;3437  Q_0,SC_K[.C]
U 12AC, 0D00.003C.0180.0800.0000.12AD  ;3438  D_DAL.SC
U 12AD, 0000.003E.6580.3C00.0000.0001  ;3439  ID[SBI.ERR]_D,RETURN1
            ;3440
```

```
            ;3441   .PAGE
            ;3442   .TOC "   CLRFLT ROUTINE"
            ;3443 CLRFLT: ;=================================================================
            ;3444   ;
            ;3445   ;CLEAR THE CP FAULT BIT IN THE
            ;3446   ; FAULT REGISTER
            ;3447   ;
            ;3448   ;=================================================================
            ;3449   ;
U 12AE, 0818.0038.0180.0800.0000.12AF   ;3450  D_K[.8]
U 12AF, 0000.003C.65F8.0800.0084.72B0   ;3451  Q_0,SC_K[.10]
U 12B0, 0D00.003C.0180.0800.0000.12B1   ;3452  D_DAL.SC
U 12B1, 0000.003E.6D80.3C00.0000.0001   ;3453  ID[FAULT]_D,RETURN1
            ;3454
```

```
            ;3455  .PAGE
            ;3456  .TOC "   CLRECC ROUTINE"
            ;3457 CLRECC: ;=================================================================
            ;3458  ;
            ;3459  ;CLEAR CRD,RDS BIT IN THE
            ;3460  ;SBI ERROR REGISTER
            ;3461  ;
            ;3462  ;=================================================================
            ;3463  ;
U 12B2. 0818,0038,0D80,0800,0000,12B3  ;3464  D_K[.3]
U 12B3. 0000,003C,89F8,0800,0084,72B4  ;3465  Q_0,SC_K[.D]
U 12B4. 0D00,003C,0180,0800,0000,12B5  ;3466  D_DAL,SC
U 12B5. 0000,003E,6580,3C00,0000,0001  ;3467  ID[SBI.ERR]_D,RETURN1
            ;3468
```

```
            ;3469   .PAGE
            ;3470   .TOC "    ENBECC ROUTINE"
            ;3471 ENBECC: ;=================================================================
            ;3472   ;
            ;3473   ;ENABLE CRD,RDS INTERRUPTS
            ;3474   ;
            ;3475   ;=================================================================
            ;3476   ;
U 12B6. 0818,0038,0580,0800,0000,12B7   ;3477   D_K[.1]
U 12B7. 0000,003C,61F8,0800,0084,72B8   ;3478   Q_0,SC_K[.F]
U 12B8. 0D00,003C,0180,0800,0000,12B9   ;3479   D_DAL,SC
U 12B9. 0000,003E,6580,3C00,0000,0001   ;3480   ID[SBI.ERR]_D,RETURN1
            ;3481
            ;3482
```

```
            ;3483  .PAGE
            ;3484  .TOC "   ENBLFLT ROUTINE"
            ;3485 ;*****************************************************************
            ;3486 ;*
            ;3487 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;3488 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;3489 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;3490 ;
            ;3491 ; D AND SC GET CLOBBERED
            ;3492 ;-
            ;3493 ;*****************************************************************
U 12BA. 0F00.003C.2180.0800.0084.72BB  ;3494 ENBFLT: SC_K[.14],D_0
U 12BB. 0000.003C.0580.0800.0084.B2BC  ;3495  SC_SC-K[.1]
U 12BC. 0801.001C.0180.0800.0000.12BD  ;3496  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 12BD. 0600.003C.6D80.3C00.0000.12BE  ;3497  ID[FAULT]_D,D_D.RIGHT
U 12BE. 0000.003E.6D80.3C00.0000.0001  ;3498  ID[FAULT]_D,RETURN1
            ;3499
```

```
            ;3500  .PAGE
            ;3501  .TOC "    NOINTR ROUTINE"
            ;3502 ;*****************************************************************
            ;3503 ;*
            ;3504 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
            ;3505 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
            ;3506 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
            ;3507 ; IF ANY INTERRUPTS OCCURRED:
            ;3508 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
            ;3509 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
            ;3510 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
            ;3511 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
            ;3512 ;  RETURN1 IS USED TO GET BACK
            ;3513 ; ELSE
            ;3514 ;  RETURN2 IS USED
            ;3515 ;
            ;3516 ; THE D AND Q REGISTERS GET CLOBBERED.
            ;3517 ;
            ;3518 ;-
            ;3519 ;*****************************************************************
            ;3520 =0
U 1130. 0000.003D.0180.0800.0000.12C4  ;3521 NOINTR: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1131. 0000.003C.35F0.2C00.0000.12BF  ;3522  Q_ID[VECTOR]
U 12BF. 0019.2034.8180.0800.0010.12C0  ;3523  ALU_Q.AND.K[.3FF],CLK.UBCC
U 12C0. 0000.013C.0180.0800.0000.1132  ;3524  Z?
U 1132. 0000.003C.0180.0800.0000.12C8  ;3525 =0 J/ERRFLT   ; INTR. OCCURRED,SET FLAG,GET DATA
U 1133. 0000.003E.0180.0800.0000.0002  ;3526  RETURN2    ; NO INTR, RETURN2
            ;3527 =0
U 1134. 0000.003D.0180.0800.0000.12C4  ;3528 WRTOUT: CALL[STROBE]   ; STROBE INTR.,WAIT DURING RETURN
U 1135. 0000.003C.35F0.2C00.0000.12C1  ;3529  Q_ID[VECTOR]
U 12C1. 0819.2034.8180.0800.0000.12C2  ;3530  ALU_Q.AND.K[.3FF],D_ALU
U 12C2. 0019.0020.A580.0800.0010.12C3  ;3531  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 12C3. 0000.013C.0180.0800.0000.1136  ;3532  Z?
U 1136. 0000.003C.0180.0800.0000.12C8  ;3533 =0 J/ERRFLT   ; WRONG VECTOR,SET FLAG,GET DATA
U 1137. 0000.003E.0180.0800.0000.0002  ;3534  RETURN2    ; OK, RETURN2
U 12C4. 0F00.003C.0180.0800.0000.12C5  ;3535 STROBE: D_0    ; SET LOWEST IPL ACTIVE
U 12C5. 0000.003C.3D80.3C00.0000.12C6  ;3536  ID[PSL]_D   ;
U 12C6. 0000.003C.0180.0800.4000.12C7  ;3537  IEK/ISTR   ; STROBE INTERRUPT
U 12C7. 0000.003E.0180.0800.0000.0001  ;3538  RETURN1    ; WAIT BEFORE LATCHING VECTOR REGISTER
U 12C8. 0018.0038.0580.09A8.0000.12C9  ;3539 ERRFLT: RC[5]_K[.1]   ; Set Flag
U 12C9. 0001.203C.65F0.2DC0.0000.12CA  ;3540  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 12CA. 0001.203C.6DF0.2DB8.0000.12CB  ;3541  RC[7]_Q,Q_ID[FAULT]  ; RC[7]_SBI.ERR
U 12CB. 0001.203E.0180.09B0.0000.0001  ;3542  RC[6]_Q,RETURN1   ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
            ;3543
```

```
;3544 .PAGE "TEST 1B3 SBI FUNCTIONAL CHECK OF VIRTUAL OPERATIONS "
;3545 ;
;3546 ;*****************************************************************************
;3547 ;++
;3548 ;
;3549 ; TEST 1AE FUNCTIONAL CHECK OF MEMORY VIRTUAL OPERATIONS
;3550 ;
;3551 ; FUNCTIONAL DESCRIPTION:
;3552 ;
;3553 ;    This test checks the following MCT functions:
;3554 ;
;3555 ;   MCT FUNCTIONS  OPERATION POINTER
;3556 ;
;3557 ;
;3558 ; SUBTEST 1 MCT/WRITE.V.NOCHK 0
;3559 ;   MCT/WRITE.V.WCHK 1
;3560 ;
;3561 ; SUBTEST 2 MCT/READ.V.RCHK  0
;3562 ;   MCT/READ.V.WCHK  1
;3563 ;   MCT/READ.V.NOCHK 2
;3564 ;
;3565 ; SUBTEST 3 LOCKREAD.V.NOCHK 0
;3566 ;   LOCKREAD.V.WCHK  1
;3567 ;
;3568 ; SUBTEST 4 LOCKWRITE.V.XCHK
;3569 ;
;3570 ; SUBTEST 5 CACHE INVALIDATE ON INTERLOCK READ
;3571 ;
;3572 ; SUBTEST 6 CACHE UPDATE ON INTERLOCK WRITE
;3573 ;
;3574 ; TB WILL BE DISABLE AND ONLY THE SBI FUNCTION
;3575 ; GENERATED WILL BE CHECK.  THE CACHE WILL BE USED TO
;3576 ; TEST TO SEE EXTENDED READS HAVE BE GENERATED.  ALSO
;3577 ; CERTAIN SEQUENCES WILL BE INITIATED TO CHECK THE
;3578 ; INTERLOCK OPERATIONS IN TERMS OF CACHE FUNCTIONALITY
;3579 ; THE STATE REGISTER IS USED TO INDEX INTO THE OPERATION TABLE
;3580 ;
;3581 ;
;3582 ;
;3583 ; ERROR LOGOUT
;3584 ;
;3585 ; SUBTESTS 1-4 ==========================
;3586 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
;3587 ;  I/O BASE ADDRESS OF CNFG C/D REGISTER
;3588 ;  UPPER EXPECTED DATA
;3589 ;  LOWER EXPECTED DATA
;3590 ;  UPPER RECEIVED DATA
;3591 ;  LOWER RECIEVED DATA
;3592 ;  OPERATION POINTER TO CONSOLE  'OR'
;3593 ;    .
;3594 ;    .
;3595 ;    .
;3596 ;
;3597 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3598 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
```

```
                ;3599 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;3600 ;  INTERRUPT FLAG:
                ;3601 ;   0=>NO INTERRUPT OCCURRED
                ;3602 ;   1=>AN INTERRUPT OCCURRED
                ;3603 ;
                ;3604 ; SUBTESTS 5-6 ======================
                ;3605 ;
                ;3606 ;   SEE INDIVIDUAL ERROR REPORTS
                ;3607 ;
                ;3608 ;
                ;3609 ;
                ;3610 ;--
                ;3611 ;*****************************************************************************
                ;3612 ;  SCRATCH PADS
                ;3613 ; RA
                ;3614 ; F U-TRAP FLAGS
                ;3615 ;
                ;3616 ;
                ;3617 ; RC
                ;3618 ; 3 TEST DATA LOWER
                ;3619 ; 4 TEST DATA UPPER
                ;3620 ; 5 OPERATION POINTER
                ;3621 ; 6 TEST ADDRESS
                ;3622 ; 8 OPERATION POINTER TO CONSOLE
                ;3623 ; 9 LOWER RECIEVED DATA
                ;3624 ; A UPPER RECIEVED DATA
                ;3625 ; B LOWER EXPECTED DATA
                ;3626 ; C UPPER EXPECTED DATA
                ;3627 ; D CODE FOR CONTROLLER UNDER TEST (I/O ADDRESS OF CNFG C/D REG)
                ;3628 ; E I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
                ;3629 ; F # ARG TO COSOLE
                ;3630 ;
                ;3631 ; NOTE:
                ;3632 ; SUBTEST 5 AND 6 USE THES REGISTERS DIFFERENTLY SEE INDIVIDUAL SUBTEST
                ;3633 ; FOR CORRECT USAGE.
                ;3634 ;
                ;3635 ;=============================================================================
                ;3636 ;
                ;3637 =0
U 1138, 0000,003D,0180,0800,0000,108F  ;3638 TST.27: NEWTST   ;
U 1139, 0000,003C,0180,0800,0000,1068  ;3639  NOP
U 1068, 0000,003D,0180,0800,0000,10F4  ;3640 =00 CALL[FIND.MS780E] ; Find all MS780-E's
                ;3641     ; ... on the SBI
U 1069, 0000,003C,0180,0800,0000,1368  ;3642  J/TEST.27.EXIT  ; No MS780-E's found exit test
U 106A, 0000,003C,0180,0800,0000,1070  ;3643  NOP
                ;3644 =
                ;3645 =00
                ;3646 RESTART.TEST.27:  ; Entry point for second controller
U 1070, 0000,003D,0180,0800,0000,1054  ;3647  CALL[START.TEST] ; Configure controllers
U 1071, 0000,003C,0180,0800,0000,1368  ;3648  J/TEST.27.EXIT  ; No more controllers or MS780-E's
U 1072, 0000,003C,0180,0800,0000,113A  ;3649  NOP
                ;3650 =
                ;3651 ;
                ;3652 ;=============================================================================
                ;3653 ;*
```

```
            ;3654 ;SUBTEST 1
            ;3655 ;  CHECK:   OPERATION POINTER
            ;3656 ;  MCT/WRITE.V.NOCHK  0
            ;3657 ;  MCT/WRITE.V.WCHK  1
            ;3658 ;-
            ;3659 ;=============================================================================
            ;3660 ;
            ;3661 ;
U 113A. 0000,003D,0180,0800,0000,1205  ;3662 =0 SUBTEST     ; START SUBTEST 1
U 113B. 0000,003C,1980,0800,1404,72CC  ;3663  STATE_K[ZERO]    ; SET OPERATION POINTER(FOR BEN)
U 12CC. 0018,0038,0580,0998,0000,12CD  ;3664  RC[03]_K[.1]    ; SET DATA BUFFER(LOW)
U 12CD. 0003,003C,0180,09A0,0000,12CE  ;3665  RC[04]_0    ; (UPPER)
U 12CE. 0003,003C,0180,09B0,0000,12CF  ;3666  RC[06]_0    ; SET  TEST ADDRESS
U 12CF. 0003,003C,0180,09A8,0000,113C  ;3667  RC[05]_0    ; SET OPERATION POINTER
U 113C. 0000,003D,0180,0800,0000,11FD  ;3668 =0 ERLOOP
U 113D. 0000,003C,0180,0800,0000,113E  ;3669  NOP
            ;3670 ;=============================================================================
            ;3671 =0
U 113E. 0000,003D,0180,0800,0000,12A3  ;3672 M17100: CALL[INIT]    ; CLEAR THE MACHINE
            ;3673       ; SET CACHE TO OFF
U 113F. 0810,0038,0180,0928,0000,12D0  ;3674  D_RC[05]    ; SET OPERATION POINTER
U 12D0. 0001,003C,0180,09C0,0000,12D1  ;3675  RC[08]_D    ; TO CONSOLE
U 12D1. 0810,0038,0180,0918,0000,12D2  ;3676  D_RC[3]     ;SET EXPECT DATA LOW
U 12D2. 0001,003C,0180,09D8,0000,12D3  ;3677  RC[0B]_D
U 12D3. 0810,0038,0180,0920,0000,12D4  ;3678  D_RC[4]     ;SET EXPECTED DATA UPPER
U 12D4. 0001,003C,0180,09E0,0000,12D5  ;3679  RC[0C]_D
U 12D5. 0010,0038,0180,0930,0200,12D6  ;3680  VA_RC[06]    ;SET ADDRESS
U 12D6. 0810,0038,0180,0958,0000,12D7  ;3681  D_RC[0B]     ;SET WRITE DATA,LOW
U 12D7. 0000,173C,0180,0800,0000,1016  ;3682  STATE0?      ;BRANCH ON OPERATION POINTER
U 1016. 0000,003C,0180,2800,0000,1140  ;3683 =110 MCT/WRITE.V.NOCHK,J/M17110 ;OPERATION POINTER =0
U 1017. 0000,003C,0180,3000,0000,1140  ;3684  MCT/WRITE.V.WCHK,J/M17110 ;OPERATION POINTER =1
            ;3685 =0
            ;3686 M17110: D_K[.3FF].RIGHT,SI/MUL*, ; D = x1FF
U 1140. 0858,0039,8300,0800,0000,1231  ;3687    CALL[WAIT.LOOP]
U 1141. 0000,003C,0180,0800,0000,1078  ;3688  NOP
U 1078. 0000,003D,0180,0800,0000,1130  ;3689 =00 CALL[NOINTR]
U 1079. 0000,003D,F580,0800,0084,703C  ;3690  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 107A. 0810,0038,0180,0963,0000,12D8  ;3691  VA_VA+4,D_RC[0C]  ;WRITE UPPER DATA
U 12D8. 0000,173C,0180,0800,0000,1026  ;3692 = STATE0?
U 1026. 0000,003C,0180,2800,0000,1142  ;3693 =110 MCT/WRITE.V.NOCHK,J/M17120 ;RC 8=0
U 1027. 0000,003C,0180,3000,0000,1142  ;3694  MCT/WRITE.V.WCHK,J/M17120 ;RC 8=1
            ;3695 =0
            ;3696 M17120: D_K[.3FF].RIGHT,SI/MUL*, ; D = x1FF
U 1142. 0858,0039,8300,0800,0000,1231  ;3697    CALL[WAIT.LOOP]
U 1143. 0010,0038,0180,0930,0200,1088  ;3698  VA_RC[6]
U 1088. 0000,003D,0180,0800,0000,1130  ;3699 =00 CALL[NOINTR]
U 1089. 0000,003D,F580,0800,0084,703C  ;3700  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 108A. 0000,003C,0180,C800,0000,108B  ;3701 'MCT/READ.P    ;READ LOW DATA
U 108B. 0001,003C,0180,09CB,0000,108C  ;3702  RC[09]_D,VA_VA+4  ;SAVE THE DATA
U 108C. 0000,003D,0180,0800,0000,1130  ;3703 =00 CALL[NOINTR]
U 108D. 0000,003D,F580,0800,0084,703C  ;3704  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 108E. 0000,003C,0180,C800,0000,12D9  ;3705  MCT/READ.P    ;READ
U 12D9. 0001,003C,0180,09D0,0000,1090  ;3706 = RC[0A]_D    ;SAVE DATA WRITTEN
U 1090. 0000,003D,0180,0800,0000,1130  ;3707 =00 CALL[NOINTR]
U 1091. 0000,003D,F580,0800,0084,703C  ;3708  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
```

```
U 1092. 0810.0038.0180.0948.0000.12DA  ;3709  D_RC[09]    ;CHECK LOW DATA
U 12DA. 0011.0020.0180.0958.0010.12DB  ;3710 = ALU_D[XOR]RC[0B],CLK.UBCC
U 12DB. 0810.0138.0180.0950.0000.1144  ;3711  Z?,D_RC[0A]
U 1144. 0000.003D.F580.0800.0084.703C  ;3712 =0 ERROR1[.A]    ;LOW DATA BAD
U 1145. 0011.0020.0180.0960.0010.12DC  ;3713  ALU_D[XOR]RC[0C],CLK.UBCC
U 12DC. 0000.013C.0180.0800.0000.1146  ;3714  Z?
U 1146. 0000.003D.F580.0800.0084.703C  ;3715 =0 ERROR1[.A]    ;UPPER DATA BAD
            ;3716   ;
            ;3717   ;CHECK FOR LAST DATA PATTERN
            ;3718   ;
U 1147. 0010.0038.01C0.0920.0010.12DD  ;3719  Q_RC[04],CLK.UBCC
U 12DD. 0000.1B3C.0180.0800.0000.1037  ;3720  ALU.N?     ;LAST PATTERN?
U 1037. 0810.0038.0180.0918.0000.12DE  ;3721 =0111 J/M17130,D_RC[3]  ;NO, SHIFT DATA
U 103F. 0810.0038.0180.0928.0000.12E1  ;3722  J/M17140,D_RC[5]  ;YES,CHECK OPERATION
            ;3723       ;POINTER
            ;3724
U 12DE. 0500.003C.0128.0800.0000.12DF  ;3725 M17130: Q_Q.LEFT,D_D.LEFT,SI/ASHL ;SHIFT THE DATA
U 12DF. 0001.003C.0180.0998.0000.12E0  ;3726  RC[3]_D
U 12E0. 0001.203C.0180.09A0.0000.113E  ;3727  RC[4]_Q,J/M17100  ;GO REPEAT LOOP
            ;3728   ;
            ;3729   ;CHECK # OF OPERATIONS DONE
            ;3730   ; RC5 =1
U 12E1. 0019.0020.0580.0800.0010.12E2  ;3731 M17140: ALU_D[XOR]K[.1],CLK.UBCC ;LAST OPERATION DONE?
U 12E2. 0000.013C.0180.0800.0000.1148  ;3732  Z?
U 1148. 0000.003C.0180.0800.1400.D2E3  ;3733 =0 J/M17150,STATE_STATE+1  ;NO DO NEXT OPERATION
U 1149. 0000.003C.0180.0800.0000.12E6  ;3734  J/M17200    ;YES GO TO NEXT SECTION
U 12E3. 0819.0014.0580.09A8.0000.12E4  ;3735 M17150: RC[05]_ALU,D_D+K[.1]
U 12E4. 0018.0038.0580.0998.0000.12E5  ;3736  RC[3]_K[.1]    ;REINITIALIZE DATA
U 12E5. 0003.003C.0180.09A0.0000.113E  ;3737  RC[4]_0,J/M17100
            ;3738
U 12E6. 0000.003C.0180.0800.0000.114A  ;3739 M17200: NOP
            ;3740 ;
            ;3741 ;=================================================================================
            ;3742 ;+
            ;3743 ;SUBTEST 2
            ;3744 ;  CHECK:   OPERATION POINTER
            ;3745 ;  READ.V.RCHK   0
            ;3746 ;  READ.V.WCHK   1
            ;3747 ;  READ.V.NOCHK   2
            ;3748 ;
            ;3749 ; NOTE: THE CACHE WILL BE USED TO ASSURE THE EXTENTED
            ;3750 ;  READ ARE PERFORM.  A BYTE READ USING THE
            ;3751 ;  ABOVE FUNCTIONS WILL BE EXECUTED AND THE
            ;3752 ;  CACHE CHECKED TO SEE THAT THE QUAD WORD WAS WRITTEN
            ;3753 ;-
            ;3754 ;=================================================================================
            ;3755 ;
            ;3756 ;
U 114A. 0000.003D.0180.0800.0000.1205  ;3757 =0 SUBTEST
U 114B. 0000.003C.1980.0800.1404.72E7  ;3758  STATE_K[ZERO]    ;INITIALIZE OPERATION
U 12E7. 0003.003C.0180.09A8.0000.12E8  ;3759  RC[05]_0    ;POINTERS
U 12E8. 0003.003C.0180.09B0.0000.12E9  ;3760  RC[06]_0    ;SET THE ADDRESS
U 12E9. 0018.0038.0580.0998.0000.12EA  ;3761 M17210: RC[03]_K[.1]    ;INITIAL DATA PATTERN LOW
U 12EA. 0003.003C.0180.09A0.0000.114C  ;3762  RC[04]_0    ;UPPER
U 114C. 0000.003D.0180.0800.0000.11FD  ;3763 =0 ERLOOP
```

```
U 114D. 0000.003C.0180.0800.0000.114E  ;3764 M17220: NOP
U 114E. 0000.003D.0180.0800.0000.12A3  ;3765 =0 CALL[INIT]
U 114F. 0810.0038.0180.0928.0000.12EB  ;3766  D_RC[05]    ;OPERATION POINTER AND
U 12EB. 0001.003C.0180.09C0.0000.12EC  ;3767  RC[08]_D    ;DATA TO CONSOLE
U 12EC. 0810.0038.0180.0918.0000.12ED  ;3768  D_RC[3]
U 12ED. 0001.003C.0180.09D8.0000.12EE  ;3769  RC[0B]_D
U 12EE. 0810.0038.0180.0920.0000.12EF  ;3770  D_RC[4]
U 12EF. 0001.003C.0180.09E0.0000.12F0  ;3771  RC[0C]_D
U 12F0. 0010.0038.0180.0930.0200.12F1  ;3772  VA_RC[06]    ;SET THE ADDRESS
U 12F1. 0810.0038.0180.0958.0000.12F2  ;3773  D_RC[0B]    ;SET THE DATA
U 12F2. 0000.003C.0180.A800.0000.12F3  ;3774  CACHE.P_D[LONG]    ;WRITE LOWER DATA
U 12F3. 0000.003C.0180.0800.0000.1150  ;3775  NOP
          ;3776 =0 D_K[.6],
U 1150. 0818.0039.D580.0800.0000.1231  ;3777   CALL[WAIT.LOOP]
U 1151. 0000.003C.0180.0800.0000.1094  ;3778  NOP
          ;3779
U 1094. 0000.003D.0180.0800.0000.1130  ;3780 =00 CALL[NOINTR]
U 1095. 0000.003D.F580.0800.0084.703C  ;3781  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1096. 0810.0038.0180.0963.0000.12F4  ;3782  VA_VA+4,D_RC[0C]
          ;3783 =
U 12F4. 0000.003C.0180.A800.0000.12F5  ;3784  CACHE.P_D[LONG]    ;WRITE UPPER DATA
U 12F5. 0000.003C.0180.0800.0000.1152  ;3785  NOP
          ;3786 =0 D_K[.6],
U 1152. 0818.0039.D580.0800.0000.1231  ;3787   CALL[WAIT.LOOP]
U 1153. 0000.003C.0180.0800.0000.1098  ;3788  NOP
U 1098. 0000.003D.0180.0800.0000.1130  ;3789 =00 CALL[NOINTR]
U 1099. 0000.003D.F580.0800.0084.703C  ;3790  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 109A. 0F00.003C.0180.0800.0000.12F6  ;3791  D_0    ;TURN CACHE ON
          ;3792 =
U 12F6. 0000.003C.7580.3C00.0000.12F7  ;3793  ID[MAINT]_D
U 12F7. 0010.0038.0180.0930.0200.12F8  ;3794  VA_RC[6]
U 12F8. 0000.003C.0180.9000.0000.12F9  ;3795  MCT/INVALIDATE    ;INVALIDATE LOWER
U 12F9. 0000.003C.0180.0803.0000.12FA  ;3796  VA_VA+4
U 12FA. 0000.003C.0180.9000.0000.12FB  ;3797  MCT/INVALIDATE    ;INVALIDATE UPPER
          ;3798    ;
          ;3799    ;DO THE OPERATION SPECIFIED
          ;3800    ;BY THE OPERATION POINTER
          ;3801    ;
U 12FB. 0010.1738.0180.0930.0200.1074  ;3802  STATE1-0?,VA_RC[06]  ;OPERATION POINTER(RC5)
U 1074. 0000.803C.0180.4000.0000.12FC  ;3803 =100 MCT/READ.V.RCHK,DT/BYTE,J/M17250 ;=0
U 1075. 0000.803C.0180.5000.0000.12FC  ;3804  MCT/READ.V.WCHK,DT/BYTE,J/M17250 ;=1
U 1076. 0000.803C.0180.4800.0000.12FC  ;3805  MCT/READ.V.NOCHK,DT/BYTE,J/M17250 ;=2
          ;3806 =
U 12FC. 0F00.003C.8580.0800.0084.72FD  ;3807 M17250: SC_K[.C],D_0    ;CACHE ON,SBI OFF
U 12FD. 0801.001C.0180.0800.0000.12FE  ;3808  D_D.ORNOT.MASK
U 12FE. 0010.0038.7580.3D30.0200.12FF  ;3809  ID[MAINT]_D,VA_RC[06]
U 12FF. 0000.003C.0180.C800.0000.109C  ;3810  MCT/READ.P    ;READ LOWER DATA
U 109C. 0001.003D.0180.09C8.0000.1130  ;3811 =00 CALL[NOINTR],RC[09]_D
U 109D. 0000.003D.F580.0800.0084.703C  ;3812  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 109E. 0818.0038.D5F8.0800.0000.1300  ;3813  D_K[.6],Q_0    ;CHECK FOR CACHE HIT
U 1300. 0000.003C.0180.0800.0084.7301  ;3814 = SC_K[.8]
U 1301. 0D00.003C.75F0.2C00.0000.1302  ;3815  D_DAL.SC,Q_ID[MAINT]
U 1302. 001D.0034.0180.0800.0010.1303  ;3816  ALU_D[AND]Q,CLK.UBCC
U 1303. 0F00.013C.0180.0803.0000.1154  ;3817  Z?,D_0,VA_VA+4
U 1154. 0000.003C.0180.0800.0000.1304  ;3818 =0 J/M17260
```

```
U 1155. 0000.003D.F580.0800.0084.703C  ;3819  ERROR1[.A]    ;NOT A CACHE HIT,LOWER
U 1304. 0000.003C.0180.C800.0000.10A0  ;3820 M17260: MCT/READ.P    ;READ UPPER DATA
U 10A0. 0001.003D.0180.09D0.0000.1130  ;3821 =00 CALL[NOINTR],RC[0A]_D  ;SAVE UPPER DATA
U 10A1. 0000.003D.F580.0800.0084.703C  ;3822  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 10A2. 0818.0038.D5F8.0800.0000.1305  ;3823  D_K[.6],Q_0    ;CHECK FOR CACHE HIT UPPER
U 1305. 0000.003C.0180.0800.0084.7306  ;3824 = SC_K[.8]
U 1306. 0D00.003C.75F0.2C00.0000.1307  ;3825  D_DAL.SC,Q_ID[MAINT]
U 1307. 001D.0034.0180.0800.0010.1308  ;3826  ALU_D[AND]Q,CLK.UBCC
U 1308. 0000.013C.0180.0800.0000.1156  ;3827  Z?
U 1156. 0810.0038.0180.0948.0000.1309  ;3828 =0 J/M17270,D_RC[09]
U 1157. 0000.003D.F580.0800.0084.703C  ;3829  ERROR1[.A]    ;NOT A CACHE HIT UPPER
U 1309. 0011.0020.0180.0958.0010.130A  ;3830 M17270: ALU_D[XOR]RC[0B],CLK.UBCC ;CHECK LOW DATA
U 130A. 0010.0138.01C0.0950.0000.1158  ;3831  Z?,Q_RC[0A]
U 1158. 0000.003D.F580.0800.0084.703C  ;3832 =0 ERROR1[.A]    ;LOW DATA BAD
U 1159. 0011.2020.0180.0960.0010.130B  ;3833  ALU_Q[XOR]RC[0C],CLK.UBCC ;CHECK UPPER DATA
U 130B. 0000.013C.0180.0800.0000.115A  ;3834  Z?
U 115A. 0000.003D.F580.0800.0084.703C  ;3835 =0 ERROR1[.A]    ;UPPER DATA BAD
            ;3836   ;
            ;3837   ;CHECK TEST DATA FOR
            ;3838   ;TEST PATTERN AND SHIFT
            ;3839   ;
U 115B. 0010.0038.01C0.0920.0010.130C  ;3840  Q_RC[04],CLK.UBCC  ;CHECK UPPER BIT 31 FOR SET
U 130C. 0000.1B3C.0180.0800.0000.1047  ;3841  ALU.M?
U 1047. 0810.0038.0180.0918.0000.130D  ;3842 =0111 J/M17280,D_RC[3]  ;GO SHIFT RC3,4
U 104F. 0810.0038.0180.0928.0000.1310  ;3843  J/M17290,D_RC[5]  ;GO CHECK OPERATION POINTER
U 130D. 0500.003C.0128.0800.0000.130E  ;3844 M17280: Q_Q.LEFT,D_D.LEFT,SI/ASHL ;SHIFT
U 130E. 0001.003C.0180.0998.0000.130F  ;3845  RC[03]_D
U 130F. 0001.203C.0180.09A0.0000.114D  ;3846  RC[04]_Q,J/M17220  ;GO REPEAT LOOP
            ;3847   ;
U 1310. 0019.0020.0980.0800.0010.1311  ;3848 M17290: ALU_D.XOR.K[.2],CLK.UBCC ;CHECK OPERATION POINTER
U 1311. 0000.013C.0180.0800.0000.115C  ;3849  Z?
U 115C. 0000.003C.0180.0800.0000.1312  ;3850 =0 J/M17295
U 115D. 0000.003C.0180.0800.0000.1314  ;3851  J/M17300    ;=3, GOTO NEXT SECTION
            ;3852 M17295: ;
            ;3853  ;INCREMENT OPERATION POINTERS
            ;3854  ;AND REPEAT TEST LOOP
            ;3855  ;
U 1312. 0819.0014.0580.09A8.0000.1313  ;3856  RC[05]_ALU,D_D+K[.1]
U 1313. 0000.003C.0180.0800.1400.D2E9  ;3857  STATE_STATE+1,J/M17210
U 1314. 0000.003C.0180.0800.0000.115E  ;3858 M17300: NOP
            ;3859 ;
            ;3860 ;================================================================================
            ;3861 ;*
            ;3862 ;SUBTEST 3
            ;3863 ;
            ;3864 ; CHECK:    OPERATION POINTER
            ;3865 ; LOCKREAD.V.NOCHK   0
            ;3866 ; LOCKREAD.V.WCHK    1
            ;3867 ;
            ;3868 ; CHECK THE VIRTUAL INTERLOCK OPERATIONS.
            ;3869 ; THE TB IS FORCED OFF ONLY, THE OPERATION
            ;3870 ; ON THE SBI IS CHECKED.  THE CACHE
            ;3871 ; WILL BE ENABLED TO ASSURE THE DATA
            ;3872 ; IS NOT CACHED.  THE CACHE WILL BE
            ;3873 ; ONLY CHECKED FOR MISS. THE LOCKWRITE.V.XCHK
```

```
            ;3874 ; OPERATION WILL BE CHECKED IN THE NEXT
            ;3875 ; SECTION.
            ;3876 ;-
            ;3877 ;================================================================================
            ;3878 ;
U 115E. 0000.003D.0180.0800.0000.1205   ;3879 =0 SUBTEST
U 115F. 0000.003C.1980.0800.1404.7315   ;3880  STATE_K[ZERO]   ;SET OPERATION POINTER
U 1315. 0003.003C.0180.09A8.0000.1316   ;3881  RC[05]_0
U 1316. 0003.003C.0180.09A0.0000.1317   ;3882  RC[04]_0    ;SET UPPER DATA
U 1317. 0003.003C.0180.09E0.0000.1318   ;3883  RC[0C]_0    ;ALL INTERLOCK OPERATIONS
U 1318. 0003.003C.0180.09D0.0000.1319   ;3884  RC[0A]_0    ;ONE LONG READS,NOT QUAD
U 1319. 0003.003C.0180.09B0.0000.131A   ;3885  RC[06]_0    ;SET TEST ADDRESS
U 131A. 0018.0038.0580.0998.0000.1160   ;3886 M17310: RC[03]_K[.1]
U 1160. 0000.003D.0180.0800.0000.11FD   ;3887 =0 ERLOOP
U 1161. 0000.003C.0180.0800.0000.1162   ;3888 M17320: NOP
U 1162. 0000.003D.0180.0800.0000.12A3   ;3889 =0 CALL[INIT]
U 1163. 0010.0038.0180.0930.0200.1164   ;3890  VA_RC[6]
            ;3891 =0 D_K[.3FF].RIGHT,SI/MUL*. ; D = x1FF
U 1164. 0858.0039.8300.0800.0000.1231   ;3892    CALL[WAIT.LOOP]
U 1165. 0810.0038.0180.0928.0000.131B   ;3893  D_RC[05]    ;GET OPERATION POINTER
U 131B. 0001.003C.0180.09C0.0000.131C   ;3894  RC[08]_D    ;TO CONSOLE
U 131C. 0810.0038.0180.0918.0000.131D   ;3895  D_RC[03]    ;SET EXPECTED DATA
U 131D. 0001.003C.0180.09D8.0000.131E   ;3896  RC[0B]_D    ;TO CONSOLE
U 131E. 0810.0038.0180.0958.0000.131F   ;3897  D_RC[0B]    ;WRITE DATA
U 131F. 0000.003C.0180.A800.0000.1166   ;3898  MCT/WRITE.P
            ;3899 =0 D_K[.6],
U 1166. 0818.0039.D580.0800.0000.1231   ;3900    CALL[WAIT.LOOP]
U 1167. 0000.003C.0180.0800.0000.10A4   ;3901  NOP
U 10A4. 0000.003D.0180.0800.0000.1130   ;3902 =00 CALL[NOINTR]
U 10A5. 0000.003D.F580.0800.0084.703C   ;3903  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 10A6. 0F10.0038.0180.0930.0200.1320   ;3904  D_0,VA_RC[06]    ;INVALIDATE THE CACHE
U 1320. 0000.003C.7580.3C00.0000.1321   ;3905 = ID[MAINT]_D    ;
U 1321. 0000.003C.0180.9000.0000.1322   ;3906  MCT/INVALIDATE
U 1322. 0000.003C.0180.0803.0000.1323   ;3907  VA_VA+4
U 1323. 0000.003C.0180.9000.0000.1324   ;3908  MCT/INVALIDATE
U 1324. 0010.1738.0180.0930.0200.107E   ;3909  STATE0?,VA_RC[06]  ;SET OPERATION POINTER AND ADDRESS
U 107E. 0000.003C.0180.6800.0000.10A8   ;3910 =110 MCT/LOCKREAD.V.NOCHK,J/M17330 ;OPERATION=0
U 107F. 0000.003C.0180.7000.0000.10A8   ;3911  MCT/LOCKREAD.V.WCHK,J/M17330 ;          =1
            ;3912 =00
U 10A8. 0001.003D.0180.09C8.0000.1130   ;3913 M17330: RC[09]_D,CALL[NOINTR]
U 10A9. 0000.003D.F580.0800.0084.703C   ;3914  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 10AA. 0818.0038.D5F8.0800.0000.1325   ;3915  D_K[.6],Q_0    ;CHECK FOR CACHE MISS
U 1325. 0000.003C.0180.0800.0084.7326   ;3916 = SC_K[.8]
U 1326. 0D00.003C.75F0.2C00.0000.1327   ;3917  D_DAL.SC,Q_ID[MAINT]
U 1327. 001D.0034.0180.0800.0010.1328   ;3918  ALU_D[AND]Q,CLK.UBCC
U 1328. 0010.0138.0180.0930.0200.1168   ;3919  Z?,VA_RC[6]
U 1168. 0000.003D.F580.0800.0084.703C   ;3920 =0 ERROR1[.A]    ;CACHE HIT
U 1169. 0000.003C.0180.C800.0000.10AC   ;3921  MCT/READ.P    ;READ AND SEE MISS,IF HIT INTERLOCK CACHED
U 10AC. 0000.003D.0180.0800.0000.1130   ;3922 =00 CALL[NOINTR]
U 10AD. 0000.003D.F580.0800.0084.703C   ;3923  ERROR1[.A]    ;UNEXPECTED INTERRUPT !!!!!!!!
U 10AE. 0818.0038.D5F8.0800.0000.1329   ;3924  D_K[.6],Q_0    ;CHECK FOR CACHE MISS
U 1329. 0000.003C.0180.0800.0084.732A   ;3925 = SC_K[.8]
U 132A. 0D00.003C.75F0.2C00.0000.132B   ;3926  D_DAL.SC,Q_ID[MAINT]
            ;3927
U 132B. 001D.0034.0180.0800.0010.132C   ;3928  ALU_D[AND]Q,CLK.UBCC
```

```
U 132C. 0810.0138.0180.0958.0000.116A  ;3929  Z?,D_RC[0B]
U 116A. 0000.003D.F580.0800.0084.703C  ;3930 =0 ERROR1[.A]   ;CACHE HIT,INTERLOCK READ WAS CACHED
U 116B. 0011.0020.0180.0948.0010.132D  ;3931  ALU_D[XOR]RC[09],CLK.UBCC ;ARE THEY EQUAL
U 132D. 0810.0138.0180.0958.0000.116C  ;3932  Z?,D_RC[0B]
U 116C. 0000.003D.F580.0800.0084.703C  ;3933 =0 ERROR1[.A]   ;INTERLOCK READ DATA FAILED
U 116D. 0000.003C.0180.B800.0000.116E  ;3934  MCT/LOCKWRITE.P   ;PERFORMED BY THE PREVIOUS OPERATION
          ;3935 =0 D_K[.3FF].RIGHT,SI/MUL+,  ; D = x1FF
U 116E. 0858.0039.8300.0800.0000.1231  ;3936   CALL[WAIT.LOOP]
U 116F. 0000.003C.0180.0800.0000.10B0  ;3937  NOP
U 10B0. 0000.003D.0180.0800.0000.1130  ;3938 =00 CALL[NOINTR]
U 10B1. 0000.003D.F580.0800.0084.703C  ;3939  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10B2. 0810.0038.0180.0918.0010.132E  ;3940  D_RC[03],CLK.UBCC  ;CHECK FOR LAST PATTERN
U 132E. 0000.1B3C.0180.0800.0000.1057  ;3941 = ALU.N?
U 1057. 0500.003C.0180.0800.0000.132F  ;3942 =0111 J/M17350,D_D.LEFT  ;SHIFT AND REPEAT LOOP
U 105F. 0810.0038.0180.0928.0000.1330  ;3943  J/M17370,D_RC[5]   ;CHECK OPERATION POINTER
U 132F. 0001.003C.0180.0998.0000.1161  ;3944 M17350: RC[03]_D,J/M17320  ;GO REPEAT LOOP
U 1330. 0019.0020.0580.0800.0010.1331  ;3945 M17370: ALU_D.XOR.K[.1],CLK.UBCC ;CHECK OPERATION POINTER
U 1331. 0000.013C.0180.0800.0000.1170  ;3946  Z?
          ;3947 =0 RC[5]_ALU,D_D+K[.1],  ;CHANGE OPERATION
U 1170. 0819.0014.0580.09A8.1400.D31A  ;3948   STATE_STATE+1,J/M17310 ;POINTER
U 1171. 0000.003C.0180.0800.0000.1172  ;3949  NOP    ;END TEST
          ;3950 ;
          ;3951 ;=============================================================================
          ;3952 ;+
          ;3953 ;SUBTEST 4
          ;3954 ; LOCKWRITE.V.XCHK
          ;3955 ;-
          ;3956 ;=============================================================================
          ;3957 ;
U 1172. 0003.003D.0180.09B0.0000.1205  ;3958 =0 SUBTEST,RC[6]_0   ;SET TEST ADDRESS
U 1173. 0000.003C.1980.0800.1404.7332  ;3959  STATE_K[ZERO]   ;SET OPERATION COUNTER
U 1332. 0003.003C.0180.09A8.0000.1333  ;3960  RC[5]_0    ;TO ZERO
U 1333. 0003.003C.0180.09E0.0000.1334  ;3961  RC[0C]_0   ;SET EXPECTED AND
U 1334. 0003.003C.0180.09D0.0000.1335  ;3962  RC[0A]_0   ;DATA READ EQUAL
U 1335. 0018.0038.0580.0998.0000.1174  ;3963  RC[03]_K[.1]   ;SET INITIAL DATA
U 1174. 0000.003D.0180.0800.0000.11FD  ;3964 =0 ERLOOP
U 1175. 0000.003C.0180.0800.0000.1176  ;3965 M17400: NOP
U 1176. 0000.003D.0180.0800.0000.12A3  ;3966 =0 CALL[INIT]
U 1177. 0010.0038.0180.0930.0200.1178  ;3967  VA_RC[6]
          ;3968 =0 D_K[.3FF].RIGHT,SI/MUL+,
U 1178. 0858.0039.8300.0800.0000.1231  ;3969   CALL[WAIT.LOOP]  ;LET MEMORY INTERLOCK TIMER END
U 1179. 0F03.003C.0180.09C0.0000.1336  ;3970  RC[08]_0,D_0
U 1336. 0003.003C.0180.09E0.0000.1337  ;3971  RC[0C]_0
U 1337. 0003.003C.0180.09D0.0000.1338  ;3972  RC[0A]_0
U 1338. 0001.003C.0180.09D8.0000.1339  ;3973  RC[0B]_D   ;WRITE BACKGROUND PATTERN
U 1339. 0001.003C.0180.A9D8.0000.117A  ;3974  MCT/WRITE.P,RC[0B]_D  ;WRITE BACKGROUND PATTERN
          ;3975 =0 D_K[.6],
U 117A. 0818.0039.D580.0800.0000.1231  ;3976   CALL[WAIT.LOOP]
U 117B. 0000.003C.0180.0800.0000.10B4  ;3977  NOP
U 10B4. 0000.003D.0180.0800.0000.1130  ;3978 =00 CALL[NOINTR]
U 10B5. 0000.003D.F580.0800.0084.703C  ;3979  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10B6. 0000.003C.0180.E800.0000.133A  ;3980  MCT/LOCKREAD.P   ;READ
U 133A. 0001.003C.0180.09C8.0000.10B8  ;3981 = RC[09]_D   ;SAVE DATA
U 10B8. 0000.003D.0180.0800.0000.1130  ;3982 =00 CALL[NOINTR]
U 10B9. 0000.003D.F580.0800.0084.703C  ;3983  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
```

```
U 10BA. 0010.0038.0180.0948.0010.133B  ;3984  ALU_RC[09],CLK.UBCC  ;CHECK FOR ZERO ON
U 133B. 0810.0138.0180.0918.0000.117C  ;3985 = Z?,D_RC[3]   ;DATA READ
U 117C. 0000.003D.F580.0800.0084.703C  ;3986 =0 ERROR1[.A]   ;DIDN'T READ A ZERO
U 117D. 0000.003C.0180.3800.0000.117E  ;3987  MCT/LOCKWRITE.V.XCHK  ;INTERLOCK WRITE TEST
        ;3988 =0 D_K[.3FF].RIGHT,SI/MUL*, ; D = x1FF
U 117E. 0858.0039.8300.0800.0000.1231  ;3989   CALL[WAIT.LOOP]
U 117F. 0000.003C.0180.0800.0000.10BC  ;3990  NOP
U 10BC. 0000.003D.0180.0800.0000.1130  ;3991 =00 CALL[NOINTR]
U 10BD. 0000.003D.F580.0800.0084.703C  ;3992  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10BE. 0000.003C.0180.C800.0000.133C  ;3993  MCT/READ.P   ;CHECK DATA WRITTEN
U 133C. 0001.003C.0180.09C8.0000.10C0  ;3994 = RC[09]_D   ;SAVE DATA READ
U 10C0. 0000.003D.0180.0800.0000.1130  ;3995 =00 CALL[NOINTR]
U 10C1. 0000.003D.F580.0800.0084.703C  ;3996  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10C2. 0810.0038.0180.0918.0000.10C3  ;3997  D_RC[03]   ;SET EXPECTED DATA
U 10C3. 0001.003C.0180.09D8.0000.133D  ;3998  RC[0B]_D
U 133D. 0011.0020.0180.0948.0010.133E  ;3999  ALU_D[XOR]RC[09],CLK.UBCC ;CHECK EXPECTED AGAINST
U 133E. 0000.013C.0180.0800.0000.1180  ;4000  Z?    ;DATA READ
U 1180. 0000.003D.F580.0800.0084.703C  ;4001 =0 ERROR1[.A]   ;DIDN'T WRITE PROPERLY
U 1181. 0810.0038.0180.0918.0010.133F  ;4002  D_RC[03],CLK.UBCC  ;CHECK FOR LAST PATTERN
U 133F. 0000.1B3C.0180.0800.0000.1067  ;4003  ALU.N?
U 1067. 0500.003C.0180.0800.0000.1340  ;4004 =0111 J/M17410,D_D.LEFT  ;BIT 31=0,SHIFT AND REPEAT
U 106F. 0000.003C.0180.0800.0000.1341  ;4005  J/M17420   ;END TEST
U 1340. 0001.003C.0180.0998.0000.1175  ;4006 M17410: RC[03]_D,J/M17400  ;SHIFT DATA, REPEAT LOOP
U 1341. 0000.003C.0180.0800.0000.1182  ;4007 M17420: NOP    ;END TEST
        ;4008 ;
        ;4009 ;================================================================================
        ;4010 ;*
        ;4011 ;SUBTEST 5
        ;4012 ;
        ;4013 ; CHECK CACHE OPERATOIN ON INTERLOCK READ(BOTH GROUPS)
        ;4014 ; I.E. DATA RECIEVED IS FROM MEMORY NOT THE CACHE
        ;4015 ;
        ;4016 ; ERROR LOGOUT
        ;4017 ;
        ;4018 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
        ;4019 ;  I/O BASE ADDRESS OF CNFG C/D REGISTER
        ;4020 ;  UPPER EXPECTED DATA
        ;4021 ;  DATA READ(EITHER FROM CACHE OR MEMORY)
        ;4022 ;  CACHE GROUP POINTER(0=GROUP0,1=GROUP1,SUBTEST 5 ONLY)
        ;4023 ;  UNDEFINED
        ;4024 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
        ;4025 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
        ;4026 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
        ;4027 ;  INTERRUPT FLAG:
        ;4028 ;   0=>NO INTERRUPT OCCURRED
        ;4029 ;   1=>AN INTERRUPT OCCURRED
        ;4030 ;--
        ;4031 ;================================================================================
        ;4032 ;
        ;4033 =0
U 1182. 0000.003D.0180.0800.0000.1205  ;4034 M17500: SUBTEST
U 1183. 0003.003C.0180.09D0.0000.1184  ;4035  RC[0A]_0   ;SET GROUP POINTER
U 1184. 0000.003D.0180.0800.0000.11FD  ;4036 =0 ERLOOP
U 1185. 0000.003C.0180.0800.0000.1186  ;4037 M17505: NOP
U 1186. 0000.003D.F580.0800.0084.7202  ;4038 =0 SETRCF[.A]
```

```
U 1187, 0000.003C.0180.0800.0000.1188  ;4039  NOP
U 1188, 0000.003D.0180.0800.0000.12A3  ;4040 =0 CALL[INIT]
U 1189, 0858.0038.8300.0800.0000.118A  ;4041  D_K[.3FF].RIGHT,SI/MUL*  ; D = x1FF
U 118A, 0000.003D.0180.0800.0000.1231  ;4042 =0 CALL[WAIT.LOOP]   ;LET INTERLOCK TIMER END
U 118B, 0018.0038.1980.0800.0200.1342  ;4043  VA_ALU,ALU_K[ZERO]  ;SET ADDRESS TO ZERO
U 1342, 0010.0038.0180.0950.0010.1343  ;4044  ALU_RC[0A],CLK.UBCC  ;SET GROUP UNER TEST
U 1343, 0818.0138.4580.0800.0000.118C  ;4045  Z?,D_K[.8000]
U 118C, 0600.003C.0180.0800.0000.118D  ;4046 =0 D_D.RIGHT   ;GROUP 1
U 118D, 0600.003C.0180.0800.0000.1344  ;4047  D_D.RIGHT   ;GROUP 0
U 1344, 0000.003C.7580.3C00.0000.1345  ;4048  ID[MAINT]_D   ;TURN CACHE ON
U 1345, 0000.003C.0180.9800.0000.1346  ;4049  MCT/VALIDATE   ;VALIDATE THE CACHE
U 1346, 0000.003C.0180.0803.0000.1347  ;4050  VA_VA+4    ; FOR HITS
U 1347, 0000.003C.0180.9800.0000.1348  ;4051  MCT/VALIDATE
U 1348, 0018.0038.1980.0800.0200.1349  ;4052  VA_ALU,ALU_K[ZERO]  ;RESET ADDRESS
U 1349, 0000.003C.0180.C800.0000.10C4  ;4053  MCT/READ.P   ;QUICK CHECK FOR HIT
U 10C4, 0000.003D.0180.0800.0000.1130  ;4054 =00 CALL[NOINTR]   ; FORM VALIDATE OPERATION
U 10C5, 0000.003D.F580.0800.0084.703C  ;4055  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10C6, 0818.0038.D5F8.0800.0000.10C7  ;4056  D_K[.6],Q_0
U 10C7, 0000.003C.0180.0800.0084.734A  ;4057  SC_K[.8]
U 134A, 0D00.003C.75F0.2C00.0000.134B  ;4058  D_DAL.SC,Q_ID[MAINT]
U 134B, 001D.0034.0180.0800.0010.134C  ;4059  ALU_D[AND]Q,CLK.UBCC  ;CHECK FOR HIT
U 134C, 0000.013C.0180.0800.0000.118E  ;4060  Z?
U 118E, 0F00.003C.0180.0800.0000.134D  ;4061 =0 J/M17510,D_0
U 118F, 0000.003D.F580.0800.0084.703C  ;4062  ERROR1[.A]   ;NO HIT, VALIDATE DIDN'T WORK
          ;4063 M17510: ;
          ;4064    ;GET ALL ONES IN MEMORY LOC 0
          ;4065    ;GET ALL ZEROS IN CACHE LOCATION CORRESPONDING TO
          ;4066    ;  MEMORY LOCATION ZERO
          ;4067    ;
          ;4068    ;INTERLOCK READ SEE WHAT DATA GETS READ
          ;4069    ; IF ZEROS THEN CACHE WAS READ
          ;4070    ; IF ONES MEMORY WAS READ
          ;4071    ;
U 134D, 0801.0028.0180.0800.0000.134E  ;4072  D_NOT.D
U 134E, 0001.003C.0180.A9C8.0000.134F  ;4073  MCT/WRITE.P,RC[09]_D  ;CACHE AND MEMORY TO ALL  ONES
U 134F, 0F00.003C.8580.0800.0084.7350  ;4074  SC_K[.C],D_0   ;GENERATE CONSTANT TO DISABLE
          ;4075       ;   SBI CYCLES
U 1350, 0801.001C.0180.0800.0000.1351  ;4076  D_D.ORNOT.MASK
U 1351, 0F00.003C.7580.3C00.0000.1352  ;4077  ID[MAINT]_D,D_0   ;CACHE ON,SBI OFF
U 1352, 0000.003C.0180.A800.0000.1190  ;4078  MCT/WRITE.P
          ;4079 =0 D_K[.6],
U 1190, 0818.0039.D580.0800.0000.1231  ;4080   CALL[WAIT.LOOP]
U 1191, 0F00.003C.0180.0800.0000.1353  ;4081  D_0    ;CACHE ON, SBI ON
U 1353, 0000.003C.7580.3C00.0000.1354  ;4082  ID[MAINT]_D
U 1354, 0000.003C.0180.E800.0000.10C8  ;4083  MCT/LOCKREAD.P   ;INTERLOCK READ OPERATION
U 10C8, 0001.003D.0180.09D0.0000.1130  ;4084 =00 RC[0A]_D,CALL[NOINTR]  ;SAVE DATA READ
U 10C9, 0000.003D.F580.0800.0084.703C  ;4085  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10CA, 0810.0038.0180.0948.0000.10CB  ;4086  D_RC[09]   ;CHECK FOR EXPECTED DATA
U 10CB, 0011.0020.0180.0950.0010.1355  ;4087  ALU_D[XOR]RC[0A],CLK.UBCC
U 1355, 0000.013C.0180.0800.0000.1192  ;4088  Z?
U 1192, 0000.003D.F580.0800.0084.703C  ;4089 =0 ERROR1[.A]   ;DATA WAS BAD,I.E. CACHE READ
          ;4090       ; NOT MEMORY
U 1193, 0010.0038.0180.0950.0010.1356  ;4091  ALU_RC[0A],CLK.UBCC  ;CHECK GROUP FLAG
U 1356, 0000.013C.0180.0800.0000.1194  ;4092  Z?
U 1194, 0000.003C.0180.0800.0000.1196  ;4093 =0 J/M17600   ;IT IS SET END SUBTEST
```

```
U 1195, 0018,0038,0580,09E0,0000,1185  ;4094  RC[0C]_K[.1],J/M17505  ;SET AND REPEAT TEST
            ;4095 ;
            ;4096 ;=================================================================================
            ;4097 ;*
            ;4098 ;SUBTEST 6
            ;4099 ;
            ;4100 ; CACHE UPDATE ON AN INTERLOCK
            ;4101 ; WRITE HIT, SEE AN INTERLOCK SEQUENCE
            ;4102 ; FAULT ALSO.
            ;4103 ;
            ;4104 ; ERROR LOGOUT
            ;4105 ;
            ;4106 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780-E
            ;4107 ;  I/O BASE ADDRESS OF CNFG C/D REGISTER
            ;4108 ;  UPPER EXPECTED DATA
            ;4109 ;  DATA READ(EITHER FROM CACHE OR MEMORY)
            ;4110 ;  UNDEFINED
            ;4111 ;  UNDEFINED
            ;4112 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;4113 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;4114 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;4115 ;  INTERRUPT FLAG:
            ;4116 ;   0=>NO INTERRUPT OCCURRED
            ;4117 ;   1=>AN INTERRUPT OCCURRED;-
            ;4118 ;=================================================================================
            ;4119 ;
            ;4120 =0
U 1196, 0000,003D,0180,0800,0000,1205  ;4121 M17600: SUBTEST
U 1197, 0000,003C,0180,0800,0000,1198  ;4122  NOP
U 1198, 0000,003D,0180,0800,0000,11FD  ;4123 =0 ERLOOP
U 1199, 0F00,003C,0180,0800,0000,119A  ;4124  D_0
U 119A, 0801,0029,F580,0800,0084,7202  ;4125 =0 D_NOT.D,SETRCF[.A]
U 119B, 0001,003C,0180,09C8,0000,1357  ;4126  RC[09]_D
U 1357, 0F18,0038,1980,0800,0200,1358  ;4127  VA_ALU,ALU_K[ZERO],D_0  ;VALIDATE THE CACHE TO ZERO
U 1358, 0000,003C,7580,3C00,0000,1359  ;4128  ID[MAINT]_D   ;TURN CACHE ON
U 1359, 0000,003C,0180,9800,0000,135A  ;4129  MCT/VALIDATE
U 135A, 0000,003C,0180,0803,0000,135B  ;4130  VA_VA+4
U 135B, 0000,003C,0180,9800,0000,119C  ;4131  MCT/VALIDATE
            ;4132 =0 D_K[.3FF].RIGHT,SI/MUL+,
U 119C, 0858,0039,8300,0800,0000,1231  ;4133   CALL[WAIT.LOOP]  ; WAIT FOR INTERLOCK FLOP TO CLEAR
U 119D, 0F18,0038,1980,0800,0200,135C  ;4134  D_0,VA_ALU,ALU_K[ZERO]
U 135C, 0000,003C,0180,A800,0000,119E  ;4135  MCT/WRITE.P   ;WRITE LOWER CACHE TO ZERO
            ;4136 =0 D_K[.6],
U 119E, 0818,0039,D580,0800,0000,1231  ;4137   CALL[WAIT.LOOP]  ;
U 119F, 0000,003C,0180,0800,0000,10CC  ;4138  NOP
U 10CC, 0000,003D,0180,0800,0000,1130  ;4139 =00 CALL[NOINTR]
U 10CD, 0000,003D,F580,0800,0084,703C  ;4140  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10CE, 0810,0038,0180,0948,0000,135D  ;4141  D_RC[09]   ;SET DATA FOR INTERLOCK
U 135D, 0000,003C,0180,B800,0000,11A0  ;4142 = MCT/LOCKWRITE.P   ;WRITE
            ;4143 =0 D_K[.3FF].RIGHT,SI/MUL+,
U 11A0, 0858,0039,8300,0800,0000,1231  ;4144   CALL[WAIT.LOOP]
U 11A1, 0000,003C,0180,0800,0000,10D0  ;4145  NOP
U 10D0, 0000,003D,0180,0800,0000,1134  ;4146 =00 CALL[WRTOUT]   ;SHOULD SEE TIME OUT
U 10D1, 0000,003D,F580,0800,0084,703C  ;4147  ERROR1[.A]   ;I.E. MEMORY DIDN'T RESPOND
U 10D2, 0000,003C,0180,0800,0000,135E  ;4148  NOP    ;TO INTERLOCK FAULT
```

```
U 135E. 0000.003C.0180.0800.0000.11A2  ;4149 = NOP    ;
U 11A2. 0000.003D.0180.0800.0000.12A9  ;4150 =0 CALL[CLRTIM]   ;CLEAR TIMEOUT BIT
U 11A3. 0000.003C.0180.0800.0000.11A4  ;4151  NOP
U 11A4. 0000.003D.0180.0800.0000.12AE  ;4152 =0 CALL[CLRFLT]   ;CLEAR CPU FAULT
U 11A5. 0000.003C.0180.0800.0000.11A6  ;4153  NOP
U 11A6. 0000.003D.0180.0800.0000.12BA  ;4154 =0 CALL[ENBFLT]   ;ENABLE CPU FAULT
U 11A7. 0F00.003C.8980.0800.0084.735F  ;4155  SC_K[.D],D_0   ;CHECK FOR HIT AND UPDATED DATA
U 135F. 0801.001C.0180.0800.0000.1360  ;4156  D_D.ORNOT.MASK
U 1360. 0000.003C.7580.3C00.0000.1361  ;4157  ID[MAINT]_D   ;SBI OFF,CACHE ON
U 1361. 0000.003C.0180.C800.0000.10D4  ;4158  MCT/READ.P   ;READ
U 10D4. 0001.003D.0180.09D0.0000.1130  ;4159 =00 RC[0A]_D,CALL[NOINTR]  ;SAVE THE DATA
U 10D5. 0000.003D.F580.0800.0084.703C  ;4160  ERROR1[.A]   ;UNEXPECTED INTERRUPT !!!!!!!!
U 10D6. 0818.0038.D5F8.0800.0000.1362  ;4161  D_K[.6],Q_0   ;CHECK FOR CACHE HIT
U 1362. 0000.003C.0180.0800.0084.7363  ;4162 = SC_K[.8]
U 1363. 0D00.003C.75F0.2C00.0000.1364  ;4163  D_DAL.SC,Q_ID[MAINT]
U 1364. 001D.0034.0180.0800.0010.1365  ;4164  ALU_D[AND]Q,CLK.UBCC
U 1365. 0000.013C.0180.0800.0000.11A8  ;4165  Z?
U 11A8. 0810.0038.0180.0948.0000.1366  ;4166 =0 J/M17610,D_RC[09]  ;CACHE HIT, CHECK DATA
U 11A9. 0000.003D.F580.0800.0084.703C  ;4167  ERROR1[.A]   ;NOT A CACHE HIT
U 1366. 0011.0020.0180.0950.0010.1367  ;4168 M17610: ALU_D[XOR]RC[0A],CLK.UBCC
U 1367. 0000.013C.0180.0800.0000.11AA  ;4169  Z?
U 11AA. 0000.003D.F580.0800.0084.703C  ;4170 =0 ERROR1[.A]   ;CACHE DIDN'T UPDATE PROPERLY
U 11AB. 0000.003C.0180.0800.0000.1070  ;4171  J/RESTART.TEST.27
            ;4172 TEST.27.EXIT:
U 1368. 0000.003C.0180.0800.0000.11AC  ;4173  NOP    ; TEST FINISHED
            ;4174
            ;4175
```

```
;4176  .PAGE "TEST 1B4 INTERLEAVE MEMORY SIZING TEST"
;4177 ;*************************************************************
;4178 ;+++
;4179 ;
;4180 ; Functional Description:
;4181 ; ---------------------
;4182 ;
;4183 ; This sequence of subtests is intended to check the operation of the
;4184 ; memory sizing function of the SBI Interface.  It places the controllers
;4185 ; in all possible interleave modes and checks how the memory size field
;4186 ; changes (bits <14:9>) in CNFG register A).
;4187 ;
;4188 ;   [Subtest 1:]
;4189 ;
;4190 ; This subtest starts out by finding whether there are two controllers
;4191 ; present.  If there are, they are set to Internal Interleave Mode and
;4192 ; tests are made on the misconfiguration bits (CNFG A - bits <17:15>) and
;4193 ; the memory size bits (CNFG A - bits <14:09>).  After the total size
;4194 ; of the memory is found in interleave mode, the two controllers are set
;4195 ; to Non-Interleaved mode and a check is made to see that the size of the
;4196 ; memory available to each controller is half the total.
;4197 ;
;4198 ;   [Subtest 2:]
;4199 ;
;4200 ; In this subtest the lower controller is set to External Interleaving
;4201 ; and a check is made of the misconfiguration and size bits in CNFG A
;4202 ; (no misconfiguration is expected and teh size should be double the
;4203 ; non-interleaved size).  If the lower controller is not present this
;4204 ; subtest is skipped.
;4205 ;
;4206 ;   [Subtest 3:]
;4207 ;
;4208 ; This subtest is a repeat of the previous one, except that this time the
;4209 ; upper controller is set to External Interleaving.  If no upper
;4210 ; controller is present this subtest is skipped.
;4211 ;
;4212 ; -----------------
;4213 ;
;4214 ;
;4215 ; Error Logout:
;4216 ; ------------
;4217 ;
;4218 ; See individual error reports.
;4219 ;
;4220 ;=============================================================
;4221 ;
;4222 ; Register Map:
;4223 ; ------------
;4224 ;
;4225 ;     RA,RB     Description:
;4226 ;     -----     -----------
;4227 ;
;4228 ;       0       Holding Register for Lower Controller Memory Size
;4229 ;
;4230 ;       1       Holding Register for Upper Controller Memory Size
```

```
            ;4231 ;
            ;4232 ;       2 Controller(s) Present Flag (1 = Both present, 0 = One present)
            ;4233 ;
            ;4234 ;       3 Temporary Storage for Configuration Register A
            ;4235 ;
            ;4236 ;      RC       Description:
            ;4237 ;      --       -----------
            ;4238 ;
            ;4239 ; E I/O Base Address of MS780-E Currently Under Test
            ;4240 ;
            ;4241 ; D Configuration Register A Holding Register
            ;4242 ;
            ;4243 ; C Storage for Holding Total Memory Size
            ;4244 ;
            ;4245 ; B Lower Controller Memory Size
            ;4246 ;
            ;4247 ; A Upper Controller Memory Size
            ;4248 ;
            ;4249 ; 9 Copy of Controller(s) Present Flag
            ;4250 ;
            ;4251 ; 8 Holding Register for SBI Error Register
            ;4252 ;
            ;4253 ;--
            ;4254 ;*************************************************************
            ;4255 =0
U 11AC, 0000,003D,0180,0800,0000,108F  ;4256 T.28: NEWTST   ; Signify start of new test
U 11AD, 0000,003C,0180,0800,0000,10D8  ;4257  NOP
U 10D8, 0000,003D,0180,0800,0000,10F4  ;4258 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 10D9, 0000,003C,0180,0800,0000,10EF  ;4259  J/T.28.EXIT  ; No MS780-E's found - exit
            ;4260  ;
            ;4261  ; This entry point in this test is where we restart everything to test
            ;4262  ; the next MS780-E on the system.  If there is only one MS780 on the
            ;4263  ; system then the code will never come back to here.
            ;4264  ;
            ;4265 RESTART.TEST.28:
U 10DA, 0003,003C,0180,0A80,0000,10DB  ;4266  R[0]_0   ; Clear register to hold memory size for lower
U 10DB, 0003,003C,0180,0A88,0000,1080  ;4267  R[1]_0   ; and upper controllers
            ;4268  ;
            ;4269  ; This subroutine (FIND.CNTRLRS) is used to determine the configuration
            ;4270  ; of controller(s) on the MS780-E that is currently being tested.
            ;4271  ; There are 4 cases that are a result of this subroutine call.  These
            ;4272  ; cases are:
            ;4273  ;
            ;4274  ; CASE 1 No controllers found on this MS780-E
            ;4275  ; CASE 2 Only the lower controller was found on this MS780-E
            ;4276  ; CASE 3 Only the upper controller was found on this MS780-E
            ;4277  ; CASE 4 Both the upper and lower controller were found on this
            ;4278  ;  MS780-E
            ;4279  ;
            ;4280  ; This subroutine will also return the memory size for each controller.
            ;4281  ; The memory size for the lower controller is returned in R[0] and the
            ;4282  ; memory size for the upper controller is returned in R[1].
            ;4283  ;
U 1080, 0000,003D,0180,0800,0000,1040  ;4284 =000 CALL[FIND.CNTRLRS] ; Determine what controllers are on this MS780
U 1081, 0000,003C,0180,0800,0000,11AE  ;4285  J/T.28.NO.CTRL  ; CASE 1 - Return here if there were no
```

```
            ;4286     ; controllers found.  Call an error.
U 1082, 0000,003C,0180,0800,0000,11B0  ;4287  J/T.28.LOW.ONLY  ; CASE 2 - Return here if only the lower
            ;4288     ; controller was found.
U 1083, 0000,003C,0180,0800,0000,11B2  ;4289  J/T.28.UP.ONLY  ; CASE 3 - Return here if only the upper
            ;4290     ; controller was found.
U 1084, 0000,003C,0180,0800,0000,136D  ;4291  J/T.28.BOTH.CTRL ; CASE 4 - Return here if both controllers
            ;4292     ; were found.
            ;4293 =
```

```
            ;4294  .PAGE
            ;4295 ;
            ;4296 ;////////////////////////////////////////////////////////////////////////////////
            ;4297 ;
            ;4298 ; This error call is executed when there were no controllers found from the
            ;4299 ; call to FIND.CNTRLRS.  The test is NOT restartable after this error.
            ;4300 ;
            ;4301 =0
            ;4302 T.28.NO.CTRL:
U 11AE, 0000,003D,0580,0800,0084,703C  ;4303  ERROR1[.1]
U 11AF, 0000,003C,0180,0800,0000,11B0  ;4304  NOP
            ;4305 ;
            ;4306 ;////////////////////////////////////////////////////////////////////////////////
            ;4307 ;
            ;4308 ; ERROR LOGOUT:
            ;4309 ;
            ;4310 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4311 ;
            ;4312 ;////////////////////////////////////////////////////////////////////////////////
            ;4313 ;
            ;4314  ;
            ;4315  ; Come here when there is only a lower controller found on the MS780-E
            ;4316  ; currently under test.  Increment the Subtest counter as we go from
            ;4317  ; here directly into Subtest 2.  R0 is used as a flag to signify that
            ;4318  ; there were Both ( R0 = 1 ) or only One ( R0 = 0 ) controller(s)
            ;4319  ; found.
            ;4320  ;
            ;4321 =0
            ;4322 T.28.LOW.ONLY:
U 11B0, 0000,003D,0180,0800,0000,1205  ;4323  SUBTEST   ; Increment subtest counter as we go from here
            ;4324     ; directly to subtest 2.
U 11B1, 0003,003C,0180,0A90,0000,1369  ;4325  R[2]_0   ; Clear Flag.  This will be used in Subtests 2
            ;4326     ; and 3 to see whether both controllers present
U 1369, 0003,003C,0180,09C8,0000,136A  ;4327  RC[09]_0  ; Clear flag in save location also.
U 136A, 0000,003C,0180,0800,0000,11D0  ;4328  J/T.28.S2  ; Jump to Subtest 2
            ;4329  ;
            ;4330  ; Come here when there is only an upper controller found on the MS780-E
            ;4331  ; currently under test.  Increment the Subtest number twice as we go
            ;4332  ; directly from here to Subtest 3.
            ;4333  ;
            ;4334 =0
            ;4335 T.28.UP.ONLY:
U 11B2, 0000,003D,0180,0800,0000,1205  ;4336  SUBTEST   ; Increment the subtest counter twice
U 11B3, 0000,003C,0180,0800,0000,11B4  ;4337  NOP    ; because we are going to execute
U 11B4, 0000,003D,0180,0800,0000,1205  ;4338 =0 SUBTEST   ; Subtest 3 only.
U 11B5, 0003,003C,0180,0A90,0000,136B  ;4339  R[2]_0   ; Clear Flag.  This will be used in Subtests 2
            ;4340     ; and 3 to see whether both controllers present
U 136B, 0003,003C,0180,09C8,0000,136C  ;4341  RC[09]_0  ; Clear flag in save location also.
U 136C, 0000,003C,0180,0800,0000,11E4  ;4342  J/T.28.S3  ; Jump to Subtest 3
            ;4343  ;
            ;4344  ; Come here when both controllers were found on the MS780-E currently
            ;4345  ; under test.
            ;4346  ;
            ;4347 T.28.BOTH.CTRL:
U 136D, 0018,0038,0580,0A90,0000,136E  ;4348  R[2]_K[.1]  ; Set flag to signify that both controllers
```

```
            ;4349     ; were found.
U 136E, 0018,0038,0580,09C8,0000,136F  ;4350  RC[09]_K[.1]  ; Set flag in save location also.
U 136F, 0000,003C,0180,0800,0000,11B6  ;4351  J/T.28.S1  ; Jump to Subtest 1
            ;4352 ;
            ;4353 ;============================================================================
            ;4354 ;
            ;4355 ; Subtest 1
            ;4356 ;
            ;4357 =0
            ;4358 T.28.S1:
U 11B6, 0000,003D,0180,0800,0000,1205  ;4359  SUBTEST   ; Increment Subtest number
U 11B7, 0000,003C,0180,0800,0000,11B8  ;4360  NOP
U 11B8, 0000,003D,0180,0800,0000,11FD  ;4361 =0 ERLOOP   ; Loop on error from here
U 11B9, 0000,003C,0180,0800,0000,11BA  ;4362  NOP
U 11BA, 0000,003D,0180,0800,0000,1293  ;4363 =0 CALL[MS780E.CLEANUP] ; Clear error and diagnostic bits.
U 11BB, 0000,003C,0180,0800,0000,1370  ;4364  NOP
U 1370, 0818,0038,1180,0800,0000,10DC  ;4365  D_K[.4]   ; Move value to D to select Internal Interleave
U 10DC, 0000,003D,0180,0800,0000,1207  ;4366 =00 CALL[SELECT.CNTRLR] ; Select Internally Interleaved
U 10DD, 0000,003C,0180,0800,0000,10DE  ;4367  NOP
            ;4368  ;
            ;4369  ; This routine will read MS780 Configuration Register A and then check
            ;4370  ; the memory size on each controller against each other ( R0 is lower
            ;4371  ; memory size and R1 is upper memory size).  If they are equal then
            ;4372  ; verify memory size.  If they are Not equal then check bits 20 and
            ;4373  ; 17 in CNFG A as they should be set for this case.
            ;4374  ;
U 10DE, 0010,0038,0180,0970,0200,10DF  ;4375  VA_RC[0E]  ; Point VA to CNFG Register A
U 10DF, 0000,003C,0180,C800,0000,1371  ;4376  D[LONG]_CACHE.P  ; Read the contents of CNFB register A
U 1371, 0001,003C,0180,0A98,0000,1372  ;4377  R[3]_D   ; Save in R[3]
U 1372, 0800,003C,0180,0A08,0000,1373  ;4378  D_R[1]   ; Move upper memory size to D
U 1373, 0001,003C,0180,09D0,0000,1374  ;4379  RC[0A]_D  ; Save upper memory size in RC[0A] for error
U 1374, 0800,003C,0180,0A00,0000,1375  ;4380  D_R[0]   ; Move saved lower memory size to D
U 1375, 0001,003C,0180,09D8,0000,1376  ;4381  RC[0B]_D  ; Save lower memory size in RC[0B] for error
U 1376, 000D,0020,0180,0A08,0010,1377  ;4382  ALU_D.XOR.R[1],CLK.UBCC ; Xor two memory sizes (upper/lower)
U 1377, 0000,013C,0180,0800,0000,11BC  ;4383  Z?   ; Is the result equal to 0 ?
U 11BC, 0000,003C,0180,0800,0000,1378  ;4384 =0 J/T.28.R0.NE.R1  ; No, Unequal memory sizes on the two ctrl's
U 11BD, 0000,003C,0180,0800,0000,11CC  ;4385  J/T.28.R0.EQ.R1  ; Yes, Equal memory sizes on the two ctrl's
            ;4386  ;
            ;4387  ; Come here when R0 (lower memory size) AND R1 (upper memory size) are
            ;4388  ; not equal.
            ;4389  ;
            ;4390 T.28.R0.NE.R1:
U 1378, 0000,003C,2180,0800,0084,7379  ;4391  SC_K[.14]  ; Set-up SC to mask bit d20
U 1379, 0803,001C,0180,0800,0000,137A  ;4392  D_NOT.MASK  ; Assert bit d20 in the D register
U 137A, 0000,003C,01C0,0A18,0000,137B  ;4393  Q_R[3]   ; Move CNFG A data to the Q register
U 137B, 0001,203C,0180,09E8,0000,137C  ;4394  RC[0D]_Q  ; Save CNFG reg A data in case error
            ;4395  ;
            ;4396  ; Check bit 20 ( Error Summary ) in CNFG A
            ;4397  ;
U 137C, 001D,0034,0180,0800,0010,137D  ;4398  ALU_D.AND.Q,CLK.UBCC ; Check bit 20
U 137D, 0000,013C,0180,0800,0000,11BE  ;4399  Z?   ; Is Bit 20 asserted ?
U 11BE, 0000,003C,0180,0800,0000,137E  ;4400 =0 J/T.28.S1.CHK.17 ; Yes, Now go check if bit 17 is asserted
U 11BF, 0000,003C,0180,0800,0000,11C2  ;4401  J/T.28.S1.ERR.1  ; No, Call an error, should be set
            ;4402  ;
            ;4403  ; Bit 20 was set now check bit 17 ( Misconfiguration in Internal
```

```
            ;4404  ; Interleave Mode) in CNFG A
            ;4405  ;
            ;4406 T.28.S1.CHK.17:
U 137E, 001B.0010.6580.0800.0082.137F  ;4407  SC_K[.10]+1  ; Set-up SC to mask bit 17
U 137F, 0803.001C.0180.0800.0000.1380  ;4408  D_NOT.MASK  ; Assert bit d17 in the D register
U 1380, 001D.0034.0180.0800.0010.1381  ;4409  ALU_D.AND.Q,CLK.UBCC ; And mask value with Q to check bit 17
U 1381, 0000.013C.0180.0800.0000.11C0  ;4410  Z?  ; Was bit 17 asserted ?
U 11C0, 0000.003C.0180.0800.0000.11C4  ;4411 =0 J/T.28.S1.TRY.READ ; Yes, Now go try read to see if error conf.
U 11C1, 0000.003C.0180.0800.0000.11C2  ;4412  NOP
```

```
            ;4413  .PAGE
            ;4414 ;
            ;4415 ;////////////////////////////////////////////////////////////////////////////
            ;4416 ;
            ;4417 ; Either or Both bits 20 ( Error Summary ) and 17 ( Misconfiguration in
            ;4418 ; Internal Interleave Mode ) were not set.  Call an error.
            ;4419 ;
            ;4420 =0
            ;4421 T.28.S1.ERR.1:
U 11C2, 0000.003D,D580,0800,0084,7044  ;4422  ERROR2[.6]
U 11C3, 0000.003C,0180,0800,0000,11D0  ;4423  J/T.28.S2
            ;4424 ;
            ;4425 ;////////////////////////////////////////////////////////////////////////////
            ;4426 ;
            ;4427 ; ERROR LOGOUT:
            ;4428 ;
            ;4429 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4430 ;  Contents of Configuration Register A
            ;4431 ;  Unused
            ;4432 ;  Unused
            ;4433 ;  Unused
            ;4434 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4435 ;      0 = One Controller present
            ;4436 ;
            ;4437 ;////////////////////////////////////////////////////////////////////////////
            ;4438 ;
            ;4439  ;
            ;4440  ; Come here after determining that Bit 20 And 17 were set.  Try reading
            ;4441  ; memory location 0 which should result in a Error Confirmation sent
            ;4442  ; back to the CPU.  The error confirmation is a result of the
            ;4443  ; configuration error in the memory.
            ;4444  ;
            ;4445 =0
            ;4446 T.28.S1.TRY.READ:
U 11C4, 0000.003D,0180,0800,0000,11FD  ;4447  ERLOOP   ; Loop on error from here
U 11C5, 0000.003C,0180,0800,0000,11C6  ;4448  NOP
U 11C6, 0000.003D,0180,0800,0000,107C  ;4449 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11C7, 0000.003C,0180,0800,0000,11C8  ;4450  NOP
U 11C8, 0000.003D,8980,0800,0084,7249  ;4451 =0 SET.TRAP.MASK[.D] ; Enable trapping of time outs
U 11C9, 0003.003C,0180,0800,0200,1382  ;4452  ALU_0(A),VA_ALU  ; Point VA to location 0
U 1382, 0000.003C,0180,C800,0000,10E0  ;4453  D[LONG]_CACHE.P  ; Try reading location 0, should get error conf
U 10E0, 0000.003D,0180,0800,0084,7245  ;4454 =00 CHECK.SBI.ERR[.B] ; Check to see if error confirmation sent.
U 10E1, 0000.003C,0180,0800,0000,11CA  ;4455  J/T.28.S1.NO.ERRCNF ; Bit was not set, call an error
U 10E2, 0000.003C,0180,0800,0000,11D0  ;4456  J/T.28.S2  ; Bit was set, goto subtest 2
            ;4457 =
```

```
            ;4458  .PAGE
            ;4459 ;
            ;4460 ;////////////////////////////////////////////////////////////////////////////////
            ;4461 ;
            ;4462 ; The SBI Error register, bit 8 ( Error Confirmation ), should have been set
            ;4463 ; and it wasn't.  Call an error.
            ;4464 ;
            ;4465 =0
            ;4466 T.28.S1.NO.ERRCNF:
U 11CA. 0000.003D.5D80.0800.0084.7044  ;4467  ERROR2[.7]
U 11CB. 0000.003C.0180.0800.0000.11D0  ;4468  J/T.28.S2
            ;4469 ;
            ;4470 ;////////////////////////////////////////////////////////////////////////////////
            ;4471 ;
            ;4472 ; ERROR LOGOUT:
            ;4473 ;
            ;4474 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4475 ;  Contents of Configuration Register A
            ;4476 ;  Unused
            ;4477 ;  Unused
            ;4478 ;  Unused
            ;4479 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4480 ;       0 = One Controller present
            ;4481 ;  ID[sbi.err] Register
            ;4482 ;
            ;4483 ;////////////////////////////////////////////////////////////////////////////////
            ;4484 ;
            ;4485  ;
            ;4486  ; Come here after determining that the memory size store in R0 for the
            ;4487  ; lower and R1 for the upper is equal to each other (normal operation).
            ;4488  ; Take the total memory size and compare it to the lower memory size
            ;4489  ; multiplied by 2.  They should be equal if memory is functioning
            ;4490  ; correctly.
            ;4491  ;
            ;4492 =0
            ;4493 T.28.R0.EQ.R1:
U 11CC. 0000.003D.01B0.0800.0000.128C  ;4494  CALL[GET.MEMORY.SIZE] ; Get the memory size in Internally Interleaved
            ;4495      ; mode
U 11CD. 0001.003C.01B0.09E0.0000.1383  ;4496  RC[0C]_D  ; Save memory size in RC[0C]
U 1383. 0800.003C.01B0.0A00.0000.1384  ;4497  D_R[0]   ; Load D with the saved lower memory size
U 1384. 0021.003C.01B0.09E8.0000.1385  ;4498  RC[0D]_D.LEFT  ; Multiple lower memory size by 2 and save
U 1385. 0010.0038.01C0.0960.0000.1386  ;4499  Q_RC[0C]  ; Set-up for comperison of two sizes to see if
U 1386. 0810.0038.01B0.0968.0000.1387  ;4500  D_RC[0D]  ; they are equal - should be
U 1387. 001D.2020.01B0.0800.0010.1388  ;4501  ALU_Q.XOR.D,CLK.UBCC ; Exclusive Or the two sizes
U 1388. 0000.013C.01B0.0800.0000.11CE  ;4502  Z?   ; Are they equal ?
```

```
            ;4503  .PAGE
            ;4504 ;
            ;4505 ;////////////////////////////////////////////////////////////////////////////
            ;4506 ;
            ;4507 ; The two memory size values were not equal when they should have been.  Call
            ;4508 ; an error.
            ;4509 ;
U 11CE, 0000.003D.D580.0800.0084.7044  ;4510 =0 ERROR2[.6]
U 11CF, 0000.003C.0180.0800.0000.11D0  ;4511  NOP
            ;4512 ;
            ;4513 ;////////////////////////////////////////////////////////////////////////////
            ;4514 ;
            ;4515 ; ERROR LOGOUT:
            ;4516 ;
            ;4517 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4518 ;  Lower Controller Memory Size Multiplied by 2 (expected)
            ;4519 ;  Recieved Memory Size
            ;4520 ;  Unused
            ;4521 ;  Unused
            ;4522 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4523 ;       0 = One Controller present
            ;4524 ;
            ;4525 ;////////////////////////////////////////////////////////////////////////////
            ;4526 ;
            ;4527 ;
            ;4528 ;============================================================================
            ;4529 ;
            ;4530 ; Subtest 2
            ;4531 ;
            ;4532 =0
            ;4533 T.28.S2:
U 11D0, 0000.003D.0180.0800.0000.1205  ;4534  SUBTEST   ; Increment Subtest counter
U 11D1, 0000.003C.0180.0800.0000.11D2  ;4535  NOP
U 11D2, 0000.003D.0180.0800.0000.11FD  ;4536 =0 ERLOOP   ; Loop on error from here
U 11D3, 0000.003C.0180.0800.0000.11D4  ;4537  NOP
U 11D4, 0000.003D.0180.0800.0000.107C  ;4538 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11D5, 0000.003C.0180.0800.0000.11D6  ;4539  NOP
U 11D6, 0000.003D.0180.0800.0000.1293  ;4540 =0 CALL[MS780E.CLEANUP] ; Clear error and diagnostic bits.
U 11D7, 0000.003C.0180.0800.0000.1389  ;4541  NOP
U 1389, 0818.0038.0580.0800.0000.10E4  ;4542  D_K[.1]   ; Move value to D to select Lower controller
U 10E4, 0000.003D.0180.0800.0000.1207  ;4543 =00 CALL[SELECT.CNTRLR] ; Select Lower Controller External Interleave
U 10E5, 0000.003C.0180.0800.0000.10E7  ;4544  J/T.28.S2.MISS.CONF ; Got misconfiguration
U 10E6, 0000.003C.0180.0800.0000.11DE  ;4545  J/T.28.S2.NO.MISS.CONF ; No misconfiguration
            ;4546  ;
            ;4547  ; Come here after getting a misconfiguration return from the subroutine
            ;4548  ; SELECT.CNTRLR.  Check bit 20 and 15 in CNFG A to make sure they are
            ;4549  ; set to signify a misconfiguration.
            ;4550  ;
            ;4551 T.28.S2.MISS.CONF:
U 10E7, 0010.0038.0180.0970.0200.138A  ;4552  VA_RC[0E]  ; Got misconfiguration, Point to CNFG A
U 138A, 0000.003C.0180.C800.0000.138B  ;4553  D[LONG]_CACHE.P  ; Read CNFG register A
U 138B, 0001.003C.0180.09E0.0000.138C  ;4554  RC[0C]_D  ; Save in RC[0C]
            ;4555  ;
            ;4556  ; Check bit 20
            ;4557  ;
```

```
U 138C. 0000.003C.2180.0800.0084.738D  ;4558  SC_K[.14]  ; Set-up SC to mask bit d20
U 138D. 0803.001C.0180.0800.0000.138E  ;4559  D_NOT.MASK  ; Assert bit d20 in the D register
U 138E. 0010.0038.01C0.0960.0000.138F  ;4560  Q_RC[0C]  ; Move CNFG A data to the Q register
U 138F. 001D.0034.0180.0800.0010.1390  ;4561  ALU_D.AND.Q,CLK.UBCC ; Check bit 20
U 1390. 0000.013C.0180.0800.0000.11D8  ;4562  Z?  ; Is Bit 20 asserted ?
U 11D8. 0000.003C.0180.0800.0000.1391  ;4563 =0 J/T.28.S2.CHK.15 ; Yes, Now go check if bit 15 is asserted
U 11D9. 0000.003C.0180.0800.0000.11DC  ;4564  J/T.28.S2.ERR.1  ; No, Call an error, should be set
          ;4565  ;
          ;4566  ; Bit 20 was set now check bit 15
          ;4567  ;
          ;4568 T.28.S2.CHK.15:
U 1391. 0000.003C.6180.0800.0084.7392  ;4569  SC_K[.F]  ; Set-up SC to mask bit 15
U 1392. 0803.001C.0180.0800.0000.1393  ;4570  D_NOT.MASK  ; Assert bit d15 in the D register
U 1393. 001D.0034.0180.0800.0010.1394  ;4571  ALU_D.AND.Q,CLK.UBCC ; And mask value with Q to check bit 15
U 1394. 0000.013C.0180.0800.0000.11DA  ;4572  Z?  ; Was bit 15 asserted ?
U 11DA. 0000.003C.0180.0800.0000.11E1  ;4573 =0 J/T.28.S2.CHK.R2 ; Yes, Now go check R2 to see if both
          ;4574     ; controllers are present.
U 11DB. 0000.003C.0180.0800.0000.11DC  ;4575  NOP
```

```
            ;4576  .PAGE
            ;4577 ;
            ;4578 ;////////////////////////////////////////////////////////////////////////////
            ;4579 ;
            ;4580 ; Either or Both bits 20 ( Error Summary ) and 15 ( Misconfiguration in
            ;4581 ; Lower Controllers Memory ) were not set.  Call an error.
            ;4582 ;
            ;4583 =0
            ;4584 T.28.S2.ERR.1:
U 11DC, 0000,003D,D580,0800,0084,7044  ;4585  ERROR2[.6]
U 11DD, 0000,003C,0180,0800,0000,11E1  ;4586  J/T.28.S2.CHK.R2
            ;4587 ;
            ;4588 ;////////////////////////////////////////////////////////////////////////////
            ;4589 ;
            ;4590 ; ERROR LOGOUT:
            ;4591 ;
            ;4592 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4593 ;  Unused
            ;4594 ;  Contents of Configuration Register A
            ;4595 ;  Unused
            ;4596 ;  Unused
            ;4597 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4598 ;       0 = One Controller present
            ;4599 ;
            ;4600 ;////////////////////////////////////////////////////////////////////////////
            ;4601 ;
            ;4602 =0
            ;4603  ;
            ;4604  ; Come here when it is determined that there is no misconfiguration
            ;4605  ; returned from the SELECT.CNTRLR subroutine.  Get the total memory
            ;4606  ; size from CNFG A and compare it to the save lower memory size
            ;4607  ; multiplied by 2, they should be equal.
            ;4608  ;
            ;4609 T.28.S2.NO.MISS.CONF:
U 11DE, 0000,003D,0180,0800,0000,128C  ;4610  CALL[GET.MEMORY.SIZE] ; Get the memory size in External Interleaved
            ;4611     ; mode
U 11DF, 0001,003C,0180,09E0,0000,1395  ;4612  RC[0C]_D  ; Save memory size in RC[0C]
U 1395, 0800,003C,0180,0A00,0000,1396  ;4613  D_R[0]   ; Load D with the saved lower memory size
U 1396, 0021,003C,0180,09E8,0000,1397  ;4614  RC[0D]_D.LEFT  ; Multiple lower memory size by 2 and save
U 1397, 0010,0038,01C0,0960,0000,1398  ;4615  Q_RC[0C]  ; Set-up for comparison of two sizes to see if
U 1398, 0810,0038,0180,0968,0000,1399  ;4616  D_RC[0D]  ; they are equal - should be
U 1399, 001D,2020,0180,0800,0010,139A  ;4617  ALU_Q.XOR.D,CLK.UBCC ; Exclusive Or the two sizes
U 139A, 0000,013C,0180,0800,0000,11E0  ;4618  Z?   ; Are they equal ?
```

```
            ;4619  .PAGE
            ;4620 ;
            ;4621 ;////////////////////////////////////////////////////////////////////////////
            ;4622 ;
            ;4623 ; The two memory size values were not equal when they should have been.  Call
            ;4624 ; an error.
            ;4625 ;
U 11E0, 0000,003D,D580,0800,0084,7044  ;4626 =0 ERROR2[.6]
            ;4627 ;
            ;4628 ;////////////////////////////////////////////////////////////////////////////
            ;4629 ;
            ;4630 ; ERROR LOGOUT:
            ;4631 ;
            ;4632 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4633 ;  Lower Controller Memory Size Multiplied by 2 (expected)
            ;4634 ;  Recieved Memory Size
            ;4635 ;  Unused
            ;4636 ;  Unused
            ;4637 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4638 ;       0 = One Controller present
            ;4639 ;
            ;4640 ;////////////////////////////////////////////////////////////////////////////
            ;4641 ;
            ;4642  ;
            ;4643  ; Come here to check the value of R2 after executing subtest 2.  If
            ;4644  ; R2 is equal to 1 then goto subtest 3 as both controllers are present.
            ;4645  ; If R2 is equal to 0 then go check for another MS780-E as there was
            ;4646  ; only the lower controller present.
            ;4647  ;
            ;4648 T.28.S2.CHK.R2:
U 11E1, 0000,003C,0180,0A10,0010,139B  ;4649  ALU_R[2],CLK.UBCC ; Move R[2] ( Upper and Lower flag) to ALU
U 139B, 0000,013C,0180,0800,0000,11E2  ;4650  Z?   ; Is R[2] zero ?  Note:  if r2 = 1 then there
            ;4651     ; are two controllers, if r2 = 0 then there
            ;4652     ; are Not two controllers.
U 11E2, 0000,003C,0180,0800,0000,11E4  ;4653 =0 J/T.28.S3  ; R[2] is NOT zero, goto subtest 3
U 11E3, 0000,003C,0180,0800,0000,10EC  ;4654  J/T.28.DO.NEXT.MS780E ; R[2] is zero, restart this test
            ;4655 ;
            ;4656 ;============================================================================
            ;4657 ;
            ;4658 ; Subtest 3
            ;4659 ;
            ;4660 =0
            ;4661 T.28.S3:
U 11E4, 0000,003D,0180,0800,0000,1205  ;4662  SUBTEST   ; Increment Subtest counter
U 11E5, 0000,003C,0180,0800,0000,11E6  ;4663  NOP
U 11E6, 0000,003D,0180,0800,0000,11FD  ;4664 =0 ERLOOP   ; Loop on error from here
U 11E7, 0000,003C,0180,0800,0000,11E8  ;4665  NOP
U 11E8, 0000,003D,0180,0800,0000,107C  ;4666 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11E9, 0000,003C,0180,0800,0000,11EA  ;4667  NOP
U 11EA, 0000,003D,0180,0800,0000,1293  ;4668 =0 CALL[MS780E.CLEANUP] ; Clear error and diagnostic bits.
U 11EB, 0000,003C,0180,0800,0000,139C  ;4669  NOP
U 139C, 0818,0038,0D80,0800,0000,10E8  ;4670  D_K[.3]   ; Move value to D to select Upper controller
U 10E8, 0000,003D,0180,0800,0000,1207  ;4671 =00 CALL[SELECT.CNTRLR] ; Select Upper Controller External Interleave
U 10E9, 0000,003C,0180,0800,0000,10EB  ;4672  J/T.28.S3.MISS.CONF ; Got misconfiguration
U 10EA, 0000,003C,0180,0800,0000,11F2  ;4673  J/T.28.S3.NO.MISS.CONF ; No misconfiguration
```

```
          ;4674  ;
          ;4675  ; Come here after getting a return that specifies a misconfiguration
          ;4676  ; error in the upper controller.  Check bits 20 and 16 in CNFG A to
          ;4677  ; make sure that they are both set as they should be for a misconfig-
          ;4678  ; uration error in the upper controller.
          ;4679  ;
          ;4680 T.28.S3.MISS.CONF:
U 10EB. 0010.0038.0180.0970.0200.139D  ;4681  VA_RC[0E]  ; Got misconfiguration, Point to CNFG A
U 139D. 0000.003C.0180.C800.0000.139E  ;4682  D[LONG]_CACHE.P  ; Read CNFG register A
U 139E. 0001.003C.0180.09E0.0000.139F  ;4683  RC[0C]_D  ; Save in RC[0C]
          ;4684  ;
          ;4685  ; Check bit 20
          ;4686  ;
U 139F. 0000.003C.2180.0800.0084.73A0  ;4687  SC_K[.14]  ; Set-up SC to mask bit d20
U 13A0. 0803.001C.0180.0800.0000.13A1  ;4688  D_NOT.MASK  ; Assert bit d20 in the D register
U 13A1. 0010.0038.01C0.0960.0000.13A2  ;4689  Q_RC[0C]  ; Move CNFG A data to the Q register
U 13A2. 001D.0034.0180.0800.0010.13A3  ;4690  ALU_D.AND.Q,CLK.UBCC ; Check bit 20
U 13A3. 0000.013C.0180.0800.0000.11EC  ;4691  Z?  ; Is Bit 20 asserted ?
U 11EC. 0000.003C.0180.0800.0000.13A4  ;4692 =0 J/T.28.S3.CHK.16 ; Yes, Now go check if bit 16 is asserted
U 11ED. 0000.003C.0180.0800.0000.11F0  ;4693  J/T.28.S3.ERR.1  ; No, Call an error, should be set
          ;4694  ;
          ;4695  ; Bit 20 was set now check bit 16
          ;4696  ;
          ;4697 T.28.S3.CHK.16:
U 13A4. 0000.003C.6580.0800.0084.73A5  ;4698  SC_K[.10]  ; Set-up SC to mask bit 16
U 13A5. 0803.001C.0180.0800.0000.13A6  ;4699  D_NOT.MASK  ; Assert bit d16 in the D register
U 13A6. 001D.0034.0180.0800.0010.13A7  ;4700  ALU_D.AND.Q,CLK.UBCC ; And mask value with Q to check bit 16
U 13A7. 0000.013C.0180.0800.0000.11EE  ;4701  Z?  ; Was bit 16 asserted ?
U 11EE. 0000.003C.0180.0800.0000.10EC  ;4702 =0 J/T.28.DO.NEXT.MS780E ; Yes, Now restart this test for next MS780-E
          ;4703  ; controllers are present.
U 11EF. 0000.003C.0180.0800.0000.11F0  ;4704  NOP
```

```
            ;4705  .PAGE
            ;4706 ;
            ;4707 ;////////////////////////////////////////////////////////////////////////////////
            ;4708 ;
            ;4709 ; Either or Both bits 20 ( Error Summary ) and 16 ( Misconfiguration in
            ;4710 ; Upper Controllers Memory ) were not set.  Call an error.
            ;4711 ;
            ;4712 =0
            ;4713 T.28.S3.ERR.1:
U 11F0. 0000,003D,D580,0800,0084,7044  ;4714  ERROR2[.6]
U 11F1. 0000,003C,0180,0800,0000,10EC  ;4715  J/T.28.DO.NEXT.MS780E
            ;4716 ;
            ;4717 ;////////////////////////////////////////////////////////////////////////////////
            ;4718 ;
            ;4719 ; ERROR LOGOUT:
            ;4720 ;
            ;4721 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4722 ;  Unused
            ;4723 ;  Cnfg A Register Contents
            ;4724 ;  Unused
            ;4725 ;  Unused
            ;4726 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4727 ;       0 = One Controller present
            ;4728 ;
            ;4729 ;////////////////////////////////////////////////////////////////////////////////
            ;4730 ;
            ;4731  ;
            ;4732  ; Come here when it is determined that there is no misconfiguration
            ;4733  ; error in the upper controller.  Take the total memory size and
            ;4734  ; compare it to the upper memory size multiplied by 2, they should
            ;4735  ; be equal.
            ;4736  ;
            ;4737 =0
            ;4738 T.28.S3.NO.MISS.CONF:
U 11F2. 0000,003D,0180,0800,0000,128C  ;4739  CALL[GET.MEMORY.SIZE] ; Get the memory size in External Interleaved
            ;4740     ; mode
U 11F3. 0001,003C,0180,09E0,0000,13A8  ;4741  RC[0C]_D  ; Save memory size in RC[0C]
U 13A8. 0800,003C,0180,0A08,0000,13A9  ;4742  D_R[1]   ; Load D with the saved upper memory size
U 13A9. 0021,003C,0180,09E8,0000,13AA  ;4743  RC[0D]_D.LEFT  ; Multiple lower memory size by 2 and save
U 13AA. 0010,0038,01C0,0960,0000,13AB  ;4744  Q_RC[0C]  ; Set-up for comparison of two sizes to see if
U 13AB. 0810,0038,0180,0968,0000,13AC  ;4745  D_RC[0D]  ; they are equal - should be
U 13AC. 001D,2020,0180,0800,0010,13AD  ;4746  ALU_Q.XOR.D,CLK.UBCC ; Exclusive Or the two sizes
U 13AD. 0000,013C,0180,0800,0000,11F4  ;4747  Z?   ; Are they equal ?
```

```
            ;4748  .PAGE
            ;4749 ;
            ;4750 ;////////////////////////////////////////////////////////////////////////////////
            ;4751 ;
            ;4752 ; The two memory size values were not equal when they should have been.  Call
            ;4753 ; an error.
            ;4754 ;
U 11F4, 0000,003D,D580,0800,0084,7044  ;4755 =0 ERROR2[.6]
U 11F5, 0000,003C,0180,0800,0000,10EC  ;4756  NOP
            ;4757 ;
            ;4758 ;////////////////////////////////////////////////////////////////////////////////
            ;4759 ;
            ;4760 ; ERROR LOGOUT:
            ;4761 ;
            ;4762 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4763 ;  Lower Controller Memory Size Multiplied by 2 (expected)
            ;4764 ;  Recieved Memory Size
            ;4765 ;  Unused
            ;4766 ;  Unused
            ;4767 ;  Controller(s) Present Flag 1 = Both Controllers present
            ;4768 ;       0 = One Controller present
            ;4769 ;
            ;4770 ;////////////////////////////////////////////////////////////////////////////////
            ;4771 ;
            ;4772  ;
            ;4773  ; Come here after all possible subtest(s) have been ran on the MS780-E
            ;4774  ; that is currently under test.  Call the subroutine ENABLE.NEXT.MS780E
            ;4775  ; and if there is no more to test then go to next test else restart
            ;4776  ; this test.
            ;4777  ;
            ;4778 =00
            ;4779 T.28.DO.NEXT.MS780E:
U 10EC, 0000,003D,0180,0800,0000,127E  ;4780  CALL[ENABLE.NEXT.MS780E]
            ;4781     ; check and see if more MS780-E's ?
U 10ED, 0000,003C,0180,0800,0000,10DA  ;4782  J/RESTART.TEST.28 ; Yes, more to test, restart this test
U 10EE, 0000,003C,0180,0800,0000,10EF  ;4783  NOP   ; No, exit this test
            ;4784  ;
            ;4785  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4786  ;
            ;4787 T.28.EXIT:
U 10EF, 0000,003C,0180,0800,0000,11F6  ;4788  NOP
            ;4789
```

```
            ;4790  .PAGE "TEST 1B5 RESERVED"
            ;4791 =0
U 11F6. 0000.003D.0180.0800.0000.108F  ;4792 T1B5: NEWTST
U 11F7. 0000.003C.0180.0800.0000.11F8  ;4793  NOP
            ;4794
```

```
ZZ-ESKAR-V22  TEST 1B6 RESERVED                                                SEQ 0946
ESKAR47.MCR                    MICRO2  1P(00)    26-JUL-85  17:17:34          Page  13

            ;4795  .PAGE "TEST 1B6 RESERVED"
            ;4796 =0
U 11F8, 0000,003D,0180,0800,0000,108F  ;4797 T1B6: NEWTST
U 11F9, 0000,003C,0180,0800,0000,13AE  ;4798  NOP
            ;4799
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 0947
ESKAR47.MCR                    MICRO2  1P(00)    26-JUL-85  17:17:34          Page  13

            ;4800  .PAGE "DUMMY NEWTST"
U 13AE, 0000,003D,0180,0800,0000,108F  ;4801 DUM: NEWTST
            ;4802
```

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 0948
ESKAR47.MCR                        MICRO2  1P(00)    26-JUL-85  17:17:34       Page  13
:                                Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1929: 1908 1917= 1918= 2817= 2818= 2819= 2822=
U 1008  2417= 2419= 2421= 2426= 2267= 2268= 2269= 2270=
U 1010  2440= 2442= 2443= 2447= 1919= 1920= 3683= 3684=
U 1018  2463= 2464= 2465= 2473= 2406= 2407= 2408= 2409=
U 1020  2475= 2477= 2479= 2480= 1972= 1974= 3693= 3694=
U 1028  2974= 2975= 2976= 2977= 2978= 2979= 2980= 2981=
U 1030  2523= 2524= 2525= 2527= 1985= 1986= 1909 3721=
U 1038  2571= 2572= 2573= 2575= 2046= 2047= 1910 3722=
U 1040  3223= 3224= 3225= 3227= 2092= 2093= 1911 3842=
U 1048  3229= 3230= 3233= 3236= 2215= 2216= 2028: 3843=
U 1050  3241= 3243= 3245= 3248= 2395= 2396= 1913 3942=
U 1058  2398= 2403= 2074: 1914 2634= 2639= 2108: 3943=
U 1060  3294= 3295= 3302= 3304= 2644= 2649= 1921 4004=
U 1068  3640= 3642= 3643= 1922 2654= 2655= 1923 4005=
U 1070  3647= 3648= 3649= 1924 3803= 3804= 3805= 1925
U 1078  3689= 3690= 3691= 1926 2699= 2700= 3910= 3911=
U 1080  4284= 4285= 4287= 4289= 4291= 1927 2777= 2778=
U 1088  3699= 3700= 3701= 3702= 3703= 3704= 3705= 1961
U 1090  3707= 3708= 3709= 1962 3780= 3781= 3782= 1963
U 1098  3789= 3790= 3791= 1964 3811= 3812= 3813= 1965
U 10A0  3821= 3822= 3823= 1966 3902= 3903= 3904= 1967
U 10A8  3913= 3914= 3915= 1968 3922= 3923= 3924= 1969
U 10B0  3938= 3939= 3940= 1970 3978= 3979= 3980= 1971
U 10B8  3982= 3983= 3984= 1975 3991= 3992= 3993= 1976
U 10C0  3995= 3996= 3997= 3998= 4054= 4055= 4056= 4057=
U 10C8  4084= 4085= 4086= 4087= 4139= 4140= 4141= 1977
U 10D0  4146= 4147= 4148= 1978 4159= 4160= 4161= 1979
U 10D8  4258= 4259= 4266= 4267= 4366= 4367= 4375= 4376=
U 10E0  4454= 4455= 4456= 1980 4543= 4544= 4545= 4552=
U 10E8  4671= 4672= 4673= 4681= 4780= 4782= 4783= 4788=
U 10F0  2856= 2857= 2887= 2888= 2953= 2954= 2962= 2963=
U 10F8  2964= 2966= 2969= 2970= 3011= 3012= 3013= 3014=
U 1100  1894: 1895: 1896: 1897: 1898: 1899: 1900: 1901:
U 1108  1902: 1981 3021= 3022= 1903: 1904: 1905: 1906:
U 1110  3028= 3030= 3033= 3034= 3053= 3054= 3056= 3057=
U 1118  3146= 3147= 3150= 3151= 3162= 3163= 3166= 3167=
U 1120  3392= 3393= 3394= 3396= 3397= 3398= 3399= 3400=
U 1128  3401= 3402= 3403= 3404= 3405= 3406= 3408= 3409=
U 1130  3521= 3522= 3525= 3526= 3528= 3529= 3533= 3534=
U 1138  3638= 3639= 3662= 3663= 3668= 3669= 3672= 3674=
U 1140  3687= 3688= 3697= 3698= 3712= 3713= 3715= 3719=
U 1148  3733= 3734= 3757= 3758= 3763= 3764= 3765= 3766=
U 1150  3777= 3778= 3787= 3788= 3818= 3819= 3828= 3829=
U 1158  3832= 3833= 3835= 3840= 3850= 3851= 3879= 3880=
U 1160  3887= 3888= 3889= 3890= 3892= 3893= 3900= 3901=
U 1168  3920= 3921= 3930= 3931= 3933= 3934= 3936= 3937=
U 1170  3948= 3949= 3958= 3959= 3964= 3965= 3966= 3967=
U 1178  3969= 3970= 3976= 3977= 3986= 3987= 3989= 3990=
U 1180  4001= 4002= 4034= 4035= 4036= 4037= 4038= 4039=
U 1188  4040= 4041= 4042= 4043= 4046= 4047= 4061= 4062=
U 1190  4080= 4081= 4089= 4091= 4093= 4094= 4121= 4122=
U 1198  4123= 4124= 4125= 4126= 4133= 4134= 4137= 4138=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4144= 4145= 4150= 4151= 4152= 4153= 4154= 4155=
U 11A8  4166= 4167= 4170= 4171= 4256= 4257= 4303= 4304=
U 11B0  4323= 4325= 4336= 4337= 4338= 4339= 4359= 4360=
U 11B8  4361= 4362= 4363= 4364= 4384= 4385= 4400= 4401=
U 11C0  4411= 4412= 4422= 4423= 4447= 4448= 4449= 4450=
U 11C8  4451= 4452= 4467= 4468= 4494= 4496= 4510= 4511=
U 11D0  4534= 4535= 4536= 4537= 4538= 4539= 4540= 4541=
U 11D8  4563= 4564= 4573= 4575= 4585= 4586= 4610= 4612=
U 11E0  4626= 4649= 4653= 4654= 4662= 4663= 4664= 4665=
U 11E8  4666= 4667= 4668= 4669= 4692= 4693= 4702= 4704=
U 11F0  4714= 4715= 4739= 4741= 4755= 4756= 4792= 4793=
U 11F8  4797= 4798= 1982 1983 1984 2003 2026 2027
U 1200  2072 2073 2156 2157 2158 2176 2178 2204
U 1208  2205 2206 2207 2208 2209 2210 2211 2213
U 1210  2214 2261 2262 2263 2265 2266 2404 2405
U 1218  2430 2450 2451 2452 2459 2599 2600 2601
U 1220  2602 2603 2604 2629 2677 2701 2702 2703
U 1228  2704 2722 2723 2724 2725 2743 2744 2745
U 1230  2746 2772 2806 2808 2809 2810 2812 2813
U 1238  2814 2815 2816 2823 2824 2825 2826 2849
U 1240  2850 2851 2852 2854 2855 2882 2883 2885
U 1248  2886 2914 2915 2916 2955 2956 2957 2958
U 1250  2967 2968 2972 2973 2986 2987 2989 2990
U 1258  2991 2993 2998 2999 3000 3002 3007 3009
U 1260  3010 3018 3019 3023 3024 3025 3026 3027
U 1268  3035 3036 3037 3039 3040 3041 3042 3043
U 1270  3044 3049 3051 3052 3055 3080 3081 3082
U 1278  3083 3111 3112 3114 3115 3116 3142 3144
U 1280  3145 3149 3153 3154 3155 3156 3158 3159
U 1288  3160 3161 3164 3165 3288 3289 3290 3291
U 1290  3292 3306 3308 3354 3356 3357 3358 3360
U 1298  3362 3363 3365 3367 3368 3369 3370 3373
U 12A0  3374 3375 3378 3391 3410 3411 3423 3424
U 12A8  3425 3436 2134: 3437 3438 3439 3450 3451
U 12B0  3452 3453 3464 3465 3466 3467 3477 3478
U 12B8  3479 3480 3494 3495 3496 3497 3498 3523
U 12C0  3524 3530 3531 3532 3535 3536 3537 3538
U 12C8  3539 3540 3541 3542 3664 3665 3666 3667
U 12D0  3675 3676 3677 3678 3679 3680 3681 3682
U 12D8  3692 3706 3710 3711 3714 3720 3725 3726
U 12E0  3727 3731 3732 3735 3736 3737 3739 3759
U 12E8  3760 3761 3762 3767 3768 3769 3770 3771
U 12F0  3772 3773 3774 3775 3784 3785 3793 3794
U 12F8  3795 3796 3797 3802 3807 3808 3809 3810
U 1300  3814 3815 3816 3817 3820 3824 3825 3826
U 1308  3827 3830 3831 3834 3841 3844 3845 3846
U 1310  3848 3849 3856 3857 3858 3881 3882 3883
U 1318  3884 3885 3886 3894 3895 3896 3897 3898
U 1320  3905 3906 3907 3908 3909 3916 3917 3918
U 1328  3919 3925 3926 3928 3929 3932 3941 3944
U 1330  3945 3946 3960 3961 3962 3963 3971 3972
U 1338  3973 3974 3981 3985 3994 3999 4000 4003
U 1340  4006 4007 4044 4045 4048 4049 4050 4051
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0950
ESKAR47.MCR                    MICRO2  1P(00)    26-JUL-85  17:17:34     Page  13

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4052 4053 4058 4059 4060 4072 4073 4074
U 1350  4076 4077 4078 4082 4083 4088 4092 4127
U 1358  4128 4129 4130 4131 4135 4142 4149 4156
U 1360  4157 4158 4162 4163 4164 4165 4168 4169
U 1368  4173 4327 4328 4341 4342 4348 4350 4351
U 1370  4365 4377 4378 4379 4380 4381 4382 4383
U 1378  4391 4392 4393 4394 4398 4399 4407 4408
U 1380  4409 4410 4453 4497 4498 4499 4500 4501
U 1388  4502 4542 4553 4554 4558 4559 4560 4561
U 1390  4562 4569 4570 4571 4572 4613 4614 4615
U 1398  4616 4617 4618 4650 4670 4682 4683 4687
U 13A0  4688 4689 4690 4691 4698 4699 4700 4701
U 13A8  4742 4743 4744 4745 4746 4747 4801
U 13B0  - 13EF Unused
U 13F0  2118: 2119: 2120: 2121: 2122: 2123: 2124: 2125:
U 13F8  2126: 2127: 2128: 2129: 2130: 2131: 2132: 2133:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR47	959
Used	959
Remaining	

Total microwords used in memory U: 959
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 0952
ESKAR47.MCR                    MICRO2  1P(00)    26-JUL-85  17:17:34     Page  14


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR47.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
   ESKAR48.MCR   MICRO2 1P(00)   26-JUL-85 17 SEQ 0953

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1852 Micro Trap Handler and LSI-11 Support Routines
; 1853   Micro Trap Handler
; 1933 Micro Monitor Interface Routines
; 1934   Newtest Routine
; 1990   Error Loop Routine
; 2007   Error 1 Routine
; 2032   ERROR1 WITH PARAMETER
; 2053   Error 2 Routine
; 2079   ERROR 2 WITH PARAMETER
; 2098   Micro-Monitor Call
; 2112   Reserved Control Store
; 2139   Set Argument Count
; 2163   Increment Subtest Number
; 2184 Diagnostic Specific Subroutines
; 2185   Select Controller
; 2223   64K OR 256K CHIPS
; 2277   SELECT LOWER CONTROLLER
; 2325   SELECT UPPER CONTROLLER
; 2374   SBI Unjam Routine
; 2425   RESET SUBTEST NUMBER
; 2447   Initialize the SBI
; 2475   Enable Cache
; 2497   Disable Cache Routine
; 2519   Check for Interrupt Routine
; 2567   CHECK UTRAP
; 2599   Set Trap Mask
; 2628   FIND MS780-E MEMORY
; 2770   Generate IO Address
; 2796   Set Starting Address
; 2830   ENABLE NEXT MS780-E
; 2882   GET MEMORY SIZE
; 2941   ERDAT ROUTINE
; 2970   Set Diagnostic Mode Routine
; 2999   Check For Odd or Even TR
; 3058   REED MULLER ENCODING ROUTINE
; 3134   REED-MULLER VECTOR GENERATION ROUTINE
; 3187   FIRST ORDER REED-MULLER ENCODE
; 3268   SECOND ORDER REED-MULLER ENCODING ROUTINE
; 3359   REED-MULLER DECODE ROUTINE
; 3676   FIND CONTROLLERS
; 3756 TEST 1B7 CONTROLLER SELECTION
; 5540 TEST 1B8 RESERVED
; 5545 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR48
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; MODIFIED BY:
;1836 ;
;1837 ;  1.1 March 10, 1983  Paul Hakala
;1838 ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839 ;   the correct memory address when 256k ram chips are
;1840 ;   present.
;1841 ;
;1842 ;   Removed the LOAD.TEMP.REGS routine which isn't
;1843 ;   needed in the overlay.
;1844 ;
;1845 ;
;1846 ;
;1847 ;
;1848 ;*****************************************************************************
;1849 ;
```

```
;1850   .PAGE
;1851
;1852   .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1853   .TOC "    Micro Trap Handler"
;1854  ;++
;1855  ; FUNCTIONAL DESCRIPTION:
;1856  ;
;1857  ; This routine is responsible for 'catching' micro-traps
;1858  ; and reporting unexpected traps to the Monitor. Register R[0F]
;1859  ; contains the Trap-Expected Mask. If the specified bit is set in
;1860  ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1861  ; an error call is made.
;1862  ;
;1863  ; INPUT PARAMETERS:
;1864  ;
;1865  ; R[0F] - Expected Trap Mask
;1866  ;  Trap Code Function
;1867  ;  ---------      --------
;1868  ;       0  System Init
;1869  ;       1  Data Unaligned
;1870  ;       2  Cross Page
;1871  ;       3  TB Modify
;1872  ;       4  TB Protection
;1873  ;       6  TB Miss
;1874  ;       7  TB Parity
;1875  ;       8  Cache Parity
;1876  ;       9  Reserved Floating Operand
;1877  ;       C  Read Data Substitute
;1878  ;       D  Time out
;1879  ;       F  Control Store Parity Error
;1880  ;      10  Odd Address
;1881  ;
;1882  ; OUTPUT PARAMETERS:
;1883  ;
;1884  ; R[0F] - Mask bit is cleared.
;1885  ;
;1886  ; DATA: Micro PC of Micro Trap
;1887  ;  Trap Code (See Above)
;1888  ;  Fault Status Register
;1889  ;  SBI Error Register
;1890  ;  Timeout Address Register
;1891  ;  Maintenance Register
;1892  ;  Cache Parity Register
;1893  ;  TB Error Register 1
;1894  ;  TB Error Register 0
;1895  ;--
U 1100, 0000.003C.0180.0800.0084.7003  ;1896 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000.003C.0580.0800.0084.7001  ;1897 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000.003C.0980.0800.0084.7001  ;1898 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000.003C.0D80.0800.0084.7001  ;1899 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000.003C.1180.0800.0084.7001  ;1900 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000.003C.D580.0800.0084.7001  ;1901 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000.003C.D980.0800.0084.7001  ;1902 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000.003C.5D80.0800.0084.7001  ;1903 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000.003C.0180.0800.0084.7001  ;1904 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000.003C.8580.0800.0084.7001  ;1905 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000.003C.8980.0800.0084.7001  ;1906 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000.003C.6580.0800.0084.7001  ;1907 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000.003C.6180.0800.0084.7003  ;1908 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1909
U 1001, 0000.003C.81F0.2C00.0000.1015  ;1910 trph0: q_id[ustack]  ; Get upc of trap
U 1015, 0C00.003C.01E0.0800.0000.102A  ;1911  q_d,d_q   ; Setup to put it back on stack
U 102A, 0C00.003C.81E0.3C00.0000.102E  ;1912  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 102E, 0000.003C.01C0.0A78.0000.1043  ;1913  q_r[0f]   ; Get the Expected Trap Mask
          ;1914  alu_q.andnot.mask,
U 1043, 0001.2024.0180.0800.0010.1046  ;1915   clk.ubcc ; Was trap expected?
U 1046, 0000.013C.0180.0800.0000.1002  ;1916  z?
          ;1917 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1918       alu_q.and.mask, ; ...
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1919       return0  ; ...
U 1003, 0018.0038.1D80.09E8.0000.1016  ;1920 trph1: rc[0d]_sc  ; Output the Trap Code
U 1016, 0000.003D.D980.0800.0084.725E  ;1921 =0 setrcf[.9]  ; Set output count
U 1017, 0000.003C.49F0.2C00.0000.1053  ;1922  q_id[tber0]  ; Output all the error registers
U 1053, 0001.203C.4DF0.2DB0.0000.1056  ;1923  rc[6]_q,q_id[tber1] ; ...
U 1056, 0001.203C.65F0.2DB8.0000.105B  ;1924  rc[7]_q,q_id[sbi.err] ; ...
U 105B, 0001.203C.6DF0.2DD8.0000.1073  ;1925  rc[0b]_q,q_id[fault] ; ...
U 1073, 0001.203C.75F0.2DE0.0000.1077  ;1926  rc[0c]_q,q_id[maint] ; ...
U 1077, 0001.203C.79F0.2DC8.0000.108B  ;1927  rc[9]_q,q_id[parity] ; ...
U 108B, 0001.203C.69F0.2DC0.0000.1096  ;1928  rc[8]_q,q_id[time.eddr] ; ...
U 1096, 0001.203C.81F0.2DD0.0000.1000  ;1929  rc[0e]_q,q_id[ustack] ; ...
          ;1930 1000: ERROR1[.C],  ;
U 1000, 0001.203D.8580.09F0.0084.703C  ;1931   rc[0e]_q ; Report the error
```

```
                ;1932  .PAGE
                ;1933  .TOC " Micro Monitor Interface Routines"
                ;1934  .TOC "   Newtest Routine"
                ;1935 ;++
                ;1936 ; FUNCTIONAL DESCRIPTION:
                ;1937 ;
                ;1938 ; This routine Initializes the following registers:
                ;1939 ;  PSL    to 1F
                ;1940 ;  CES    to 0
                ;1941 ;  ACC.CS    to disable accellerator
                ;1942 ;  SIR    to 0
                ;1943 ;  CLK.CS    to stop interval timer
                ;1944 ;  TBER0    to disable the TB
                ;1945 ;  SBI.MAINT to     0
                ;1946 ;  SBI.ERR    to 0
                ;1947 ;  FAULT    to 0
                ;1948 ;  COMP    to 0
                ;1949 ;  RC[0E]    to  0
                ;1950 ;  R[0F]    to 0
                ;1951 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1952 ; A SCOPE call is then made to the Monitor.
                ;1953 ;
                ;1954 ; CALLING CONSTRAINT:
                ;1955 ;
                ;1956 ; =0
                ;1957 ;
                ;1958 ; SIDE EFFECTS:
                ;1959 ;
                ;1960 ; D Reg is destroyed
                ;1961 ; SC is destroyed
                ;1962 ;--
U 109B, 0000.003C.65F8.0800.0084.709E   ;1963 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 109E, 0818.0038.8D80.0800.0000.10BB   ;1964  d_k[.1f]  ; D=1F
U 10BB, 0D00.003C.0180.0800.0000.10BF   ;1965  d_dal.sc  ; D=001F0000
U 10BF, 0000.003C.3D80.3C00.0000.10CF   ;1966  id[psl]_d  ; psl = 001F0000
U 10CF, 0818.0038.0580.0800.0000.10D7   ;1967  d_k[.1]   ; Clear the CES register
U 10D7, 0D00.003C.0180.0800.0000.1109   ;1968  d_dal.sc  ; D = 00010000
U 1109, 0F00.003C.3180.3C00.0000.111B   ;1969  id[ces]_d,d_0  ; ces = 00010000
U 111B, 0000.003C.5D80.3C00.0000.1123   ;1970  id[acc.cs]_d  ; acc.cs = 0
U 1123, 0000.003C.3980.3C00.0000.112B   ;1971  id[sir]_d  ; sir = 0
U 112B, 0000.003C.2980.3C00.0000.124E   ;1972  id[clk.cs]_d  ; clk.cs = 0
U 124E, 0000.003C.4980.3C00.0000.1028   ;1973  id[tber0]_d  ; tber0 = 0
U 1028, 0000.003D.0180.0800.0000.1272   ;1974 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1975  intrpt.strobe,  ; Strobe interrupts
U 1029, 0818.0038.C180.0800.4000.124F   ;1976      d_k[.ffff]  ; Setup to clear SBI error register and
U 124F, 0801.0028.6580.3C00.0000.1250   ;1977  id[sbi.err]_d,d_not.d ; and fault register
U 1250, 0F00.003C.6D80.3C00.0000.1251   ;1978  id[fault]_d,d_0  ; ...
U 1251, 0000.003C.6D80.3C00.0000.1252   ;1979  id[fault]_d  ; fault = 0
U 1252, 0000.003C.7580.3C00.0000.1253   ;1980  id[maint]_d  ; MAINT = 0
U 1253, 0000.003C.6580.3C00.0000.1254   ;1981  id[sbi.err]_d  ; sbi error reg = 0
U 1254, 0000.003C.7180.3C00.0000.1255   ;1982  id[comp]_d  ; comp reg = 0
U 1255, 0000.003C.E580.3C00.0000.1256   ;1983  ID[39]_d   ; Clear code for subroutine START.TEST
U 1256, 0001.003C.0180.09F0.0000.1257   ;1984  rc[0e]_d  ;
U 1257, 0001.003C.0180.0AF8.0000.1258   ;1985  r[0f]_d   ; Clear expected trap mask
U 1258, 0001.003C.0180.09F8.0000.102C   ;1986  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0960
ESKAR48.MCR                .     MICRO2  1P(00)    26-JUL-85  17:18:58                  Page   5

U 102C, 0000,003D,0180,0800,0000,127C  ;1987 =0 call[disable.cache] ; Disable the cache
U 102D, 0F03,003C,0180,09E8,0000,105E  ;1988  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1989  .PAGE
            ;1990  .TOC "   Error Loop Routine"
            ;1991 ;**
            ;1992 ; FUNCTIONAL DESCRIPTION:
            ;1993 ;
            ;1994 ; This routine is called with the "ERLOOP" macro. The
            ;1995 ; routine calls the Micro Monitor to setup an error loop.
            ;1996 ;
            ;1997 ; CALLING CONSTRAINT:
            ;1998 ;
            ;1999 ; =0
            ;2000 ;
            ;2001 ; SIDE EFFECTS:
            ;2002 ;
            ;2003 ; D Reg - Destroyed
            ;2004 ;--
U 1259, 0818,0038,0980,0800,0000,105E  ;2005 ERLOOP: d_k[.2],j/callcon
```

```
            ;2006  .PAGE
            ;2007  .TOC "   Error 1 Routine"
            ;2008 ;**
            ;2009 ; FUNCTIONAL DESCRIPTION:
            ;2010 ;
            ;2011 ; This routine is used to report an error to the user. This
            ;2012 ; routine is used when you do not wish to continue in the
            ;2013 ; same test after the call to the Monitor.
            ;2014 ;
            ;2015 ; CALLING CONSTRAINT:
            ;2016 ;
            ;2017 ; None
            ;2018 ;
            ;2019 ; INPUTS:
            ;2020 ;
            ;2021 ; rc[0f]<15:8> - Contain the subtest number
            ;2022 ;
            ;2023 ; OUTPUTS:
            ;2024 ;
            ;2025 ; D<15:8> - Subtest number
            ;2026 ; D<7:0> - Error 1 Monitor Code
            ;2027 ;--
U 125A, 0810.0038.0180.0978.0000.125B  ;2028 ERROR1: d_rc[0f]  ; Get the subtest number
U 125B, 0819.0034.4D80.0800.0000.104E  ;2029  d_d.and.k[.ff00] ; ...
U 104E, 0819.0030.1180.0800.0000.105E  ;2030 104E: d_d.or.k[.4].j/callcon ; Call the monitor
```

```
            ;2031  .PAGE
            ;2032  .TOC "    ERROR1 WITH PARAMETER"
            ;2033 ;++
            ;2034 ; FUNCTIONAL DESCRIPTION:
            ;2035 ;
            ;2036 ; This subroutine takes  as parameter the number of arguments
            ;2037 ; to be printed to the console in the case of an error logout.
            ;2038 ;
            ;2039 ; CALLING CONSTRAINT:
            ;2040 ;
            ;2041 ; =0
            ;2042 ;INPUTS:
            ;2043 ;
            ;2044 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2045 ;--
            ;2046 =0
            ;2047 ERR1.ARG:
U 103C, 0000,003D,0180,0800,0000,125E  ;2048  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 103D, 0000,003C,0180,0800,0000,125A  ;2049  J/ERROR1  ; Go to ERROR1
            ;2050 ;
            ;2051 ;
```

```
            ;2052  .PAGE
            ;2053  .TOC "   Error 2 Routine"
            ;2054 ;++
            ;2055 ; FUNCTIONAL DESCRIPTION:
            ;2056 ;
            ;2057 ; This routine is used  to report an error to the user. This
            ;2058 ; routine is used when you wish to continue in the same test
            ;2059 ;       after the call to the Monitor.
            ;2060 ;
            ;2061 ; CALLING CONSTRAINT:
            ;2062 ;
            ;2063 ; =0
            ;2064 ;
            ;2065 ; INPUTS:
            ;2066 ;
            ;2067 ; rc[0f]<15:8> - Contain the subtest number
            ;2068 ;
            ;2069 ; OUTPUTS:
            ;2070 ;
            ;2071 ; D<15:8> - Subtest number
            ;2072 ; D<7:0> - Error 2 Monitor Code
            ;2073 ;--
U 125C, 0810,0038,0180,0978,0000,125D  ;2074 ERROR2: d_rc[0f]  ; Get the subtest number
U 125D, 0819,0034,4D80,0800,0000,105A  ;2075  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2076 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2077 ;
```

```
            ;2078  .PAGE
            ;2079  .TOC "   ERROR 2 WITH PARAMETER"
            ;2080 ;++
            ;2081 ; FUNCTIONAL DESCRIPTION:
            ;2082 ;
            ;2083 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2084 ; except that in this case after setting the number of arguments
            ;2085 ; too the console, control is transferred to routine ERROR2.
            ;2086 ;
            ;2087 ; INPUTS:
            ;2088 ;
            ;2089 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2090 ;
            ;2091 ;--
            ;2092 =0
            ;2093 ERR2.ARG:
U 1044, 0000,003D,0180,0800,0000,125E  ;2094  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1045, 0000,003C,0180,0800,0000,125C  ;2095  J/ERROR2  ; Go to ERROR2
            ;2096 ;
```

```
            ;2097  .PAGE
            ;2098  .TOC "   Micro-Monitor Call"
            ;2099 ;*
            ;2100 ; FUNCTIONAL DESCRIPTION:
            ;2101 ;
            ;2102 ; This micro word calls the micro monitor.
            ;2103 ;
            ;2104 ; EXPLICIT INPUTS:
            ;2105 ;
            ;2106 ; D reg must contain valid monitor code.
            ;2107 ;--
            ;2108 105E:
            ;2109 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2110  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2111  .PAGE
            ;2112  .TOC "   Reserved Control Store"
            ;2113 ;**
            ;2114 ; FUNCTIONAL DESCRIPTION:
            ;2115 ;
            ;2116 ; The following 16 locations must be reserved so the monitor
            ;2117 ; can use them to perform various functions that require
            ;2118 ; the execution of micro-code.
            ;2119 ;--
U 13F0, 0000,003C,0180,0B00,0000,13F1  ;2120 13F0: nop
U 13F1, 0000,003C,0180,0B00,0000,13F2  ;2121 13F1: nop
U 13F2, 0000,003C,0180,0B00,0000,13F3  ;2122 13F2: nop
U 13F3, 0000,003C,0180,0B00,0000,13F4  ;2123 13F3: nop
U 13F4, 0000,003C,0180,0B00,0000,13F5  ;2124 13F4: nop
U 13F5, 0000,003C,0180,0B00,0000,13F6  ;2125 13F5: nop
U 13F6, 0000,003C,0180,0B00,0000,13F7  ;2126 13F6: nop
U 13F7, 0000,003C,0180,0B00,0000,13F8  ;2127 13F7: nop
U 13F8, 0000,003C,0180,0B00,0000,13F9  ;2128 13F8: nop
U 13F9, 0000,003C,0180,0B00,0000,13FA  ;2129 13F9: nop
U 13FA, 0000,003C,0180,0B00,0000,13FB  ;2130 13FA: nop
U 13FB, 0000,003C,0180,0B00,0000,13FC  ;2131 13FB: nop
U 13FC, 0000,003C,0180,0B00,0000,13FD  ;2132 13FC: nop
U 13FD, 0000,003C,0180,0B00,0000,13FE  ;2133 13FD: nop
U 13FE, 0000,003C,0180,0B00,0000,13FF  ;2134 13FE: nop
U 13FF, 0000,003C,0180,0B00,0000,12AA  ;2135 13FF: nop
U 12AA, 0000,003C,0180,0B00,0000,125E  ;2136 12AA: nop ; This location is permanently blasted in the FPLA
            ;2137   ; to cause a micro ECO to location 12AA.
```

```
            ;2138  .PAGE
            ;2139  .TOC "   Set Argument Count"
            ;2140 ;++
            ;2141 ; FUNCTIONAL DESCRIPTION:
            ;2142 ;
            ;2143 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2144 ; to the console.
            ;2145 ;
            ;2146 ; CALLING CONSTRAINT:
            ;2147 ;
            ;2148 ; =0
            ;2149 ;
            ;2150 ; INPUTS:
            ;2151 ;
            ;2152 ; SC - Contains new value of argument count
            ;2153 ;
            ;2154 ; SIDE EFFECTS:
            ;2155 ;
            ;2156 ; Q is destroyed
            ;2157 ;--
U 125E, 0010,0038,01C0,0978,0000,125F  ;2158 SETRCF: q_rc[0f]  ; Get the argument count
U 125F, 0019,2034,4DC0,0800,0000,1260  ;2159  q_q.and.k[.ff00] ; ...
U 1260, 0019,2032,1D80,09F8,0000,0001  ;2160  rc[0f]_q.or.sc,returni ; Set new argument count and return
            ;2161
```

```
            ;2162  .PAGE
            ;2163  .TOC "   Increment Subtest Number"
            ;2164 ;++
            ;2165 ; FUNCTIONAL DESCRIPTION:
            ;2166 ;
            ;2167 ; This routine is used to increment the subtest number
            ;2168 ; in RC[0F]<15:8>.
            ;2169 ;
            ;2170 ; CALLING CONSTRAINT:
            ;2171 ;
            ;2172 ; =0
            ;2173 ;
            ;2174 ; SIDE EFFECTS:
            ;2175 ;
            ;2176 ; D register is destroyed
            ;2177 ;--
            ;2178 subtst:
U 1261, 0810,0038,4D80,0978,0000,1262  ;2179  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2180  rc[0f]_d+k[.ff00], ; Increment and return
U 1262, 0019,4016,4D80,09F8,0000,0001  ;2181      word,return1 ; (Subtest number is negative)
            ;2182
```

```
            ;2183  .PAGE
            ;2184  .TOC " Diagnostic Specific Subroutines"
            ;2185  .TOC "   Select Controller"
            ;2186 ;++
            ;2187 ; FUNCTIONAL DESCRIPTION:
            ;2188 ;
            ;2189 ; This routine is used to put the memory system into the
            ;2190 ; specified configuration with respect to controller select.
            ;2191 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2192 ; a RETURN2 is made.
            ;2193 ;
            ;2194 ; CALLING CONSTRAINT:
            ;2195 ;
            ;2196 ; =00
            ;2197 ;
            ;2198 ; INPUTS:
            ;2199 ;
            ;2200 ; d - Contains value to write to bits<2:0> of Register A
            ;2201 ; rc[0e] - Address of Register A
            ;2202 ;
            ;2203 ; SIDE EFFECTS:
            ;2204 ;
            ;2205 ; sc,d,va are destroyed
            ;2206 ;--
            ;2207 SELECT.CNTRLR:
U 1263, 0010,0038,0180,0970,0284,7264  ;2208  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1264, 0801,001C,0180,0800,0000,1265  ;2209  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1265, 0000,003C,0180,A800,0000,1266  ;2210  cache.p_d[long]  ; Write the configuration Register
U 1266, 0818,0038,5D80,0800,0000,1267  ;2211  d_k[.7]  ; Get Misconfiguration bits
U 1267, 0000,003C,61F8,0800,0084,7268  ;2212  sc_k[.F],q_0  ; ...
U 1268, 0D00,003C,0180,0800,0000,1269  ;2213  d_dal.sc  ; ... D = x38000
U 1269, 0000,003C,01E0,0800,0000,126A  ;2214  q_d  ; Save mask
U 126A, 0000,003C,0180,C800,0000,126B  ;2215  d[long]_cache.p  ; Read CNFG A Reg and
            ;2216  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 126B, 001D,2034,0180,0800,0010,126C  ;2217  clk.ubcc ; ...
U 126C, 0000,013C,0180,0800,0000,104C  ;2218  z?  ; Branch if no
U 104C, 0000,003E,0180,0800,0000,0001  ;2219 =0 RETURN1  ; Bad configuration
U 104D, 0000,003E,0180,0800,0000,0002  ;2220  RETURN2  ; Configuration ok
            ;2221
```

```
                ;2222  .PAGE
                ;2223  .TOC "    64K OR 256K CHIPS"
                ;2224 ;*****************************************************
                ;2225 ;**
                ;2226 ;
                ;2227 ; Functional Description:
                ;2228 ; ----------------------
                ;2229 ; This subroutine is used to find the type of chips
                ;2230 ; on the MS780-E/H arrays.  The RETURN modes are as
                ;2231 ; follows:
                ;2232 ;
                ;2233 ; RETURN1 = Error. Mixed 64K and 256K arrays
                ;2234 ;
                ;2235 ; RETURN2 = 64K chips on the array
                ;2236 ;
                ;2237 ; RETURN3 = 256K chips on the array
                ;2238 ;
                ;2239 ; Calling Constraint: =00
                ;2240 ; ------------------
                ;2241 ;
                ;2242 ;
                ;2243 ; Inputs:
                ;2244 ; ------
                ;2245 ;
                ;2246 ; RC[0E] must hold the I/O base address of the MS780-E/H
                ;2247 ; currently under test
                ;2248 ;
                ;2249 ; Outputs:
                ;2250 ; -------
                ;2251 ;
                ;2252 ; NO specific outputs. (See RETURN modes above)
                ;2253 ;
                ;2254 ;
                ;2255 ; Side Effects:
                ;2256 ; ------------
                ;2257 ;
                ;2258 ; The contents of the following registers will be lost:
                ;2259 ;
                ;2260 ; D
                ;2261 ;
                ;2262 ;--
                ;2263 ;*****************************************************
                ;2264 64K.OR.256K:
U 126D, 0010,0038,0180,0970,0200,126E  ;2265  VA_RC[0E]    ; Point to CNFG Register A
U 126E, 0000,003C,0180,C800,0000,126F  ;2266  D[LONG]_CACHE.P    ; Read the register
U 126F, 0200,003C,0180,0800,0000,1270  ;2267  D_D.RIGHT2    ; Shift received contents two bits
                ;2268       ; ...to the right
U 1270, 0600,003C,0180,0800,0000,1271  ;2269  D_D.RIGHT    ; shift D one more bit
U 1271, 0000,193C,0180,0800,0000,100C  ;2270  DI-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2271 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2272  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2273  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2274  RETURN1     ; Error: missing arrays
                ;2275
```

```
;2276  .PAGE
;2277  .TOC "    SELECT LOWER CONTROLLER"
;2278 ;*************************************************************
;2279 ;++
;2280 ;
;2281 ; Functional Description:
;2282 ; ----------------------
;2283 ;
;2284 ; This subroutine can be called to select the lower
;2285 ; Controller on a MS780-E/H.
;2286 ; If the Lower controller is misconfigured then a
;2287 ; RETURN1 will be made.  Otherwise a RETURN2
;2288 ;
;2289 ; Calling Constraint: =00
;2290 ; -----------------
;2291 ;
;2292 ;
;2293 ; Inputs:
;2294 ; ------
;2295 ;
;2296 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2297 ;
;2298 ; Outputs:
;2299 ; -------
;2300 ;
;2301 ; RC[0D] will have the address of CNFG Register C
;2302 ;        ( 2000x008 )
;2303 ;
;2304 ; Side Effects:
;2305 ; ------------
;2306 ;
;2307 ; Contents of the following registers will lost:
;2308 ;
;2309 ; D
;2310 ;
;2311 ; The Lower controller on the MS780-E/H will be configured
;2312 ; when the subroutine is exitted with a RETURN2
;2313 ;--
;2314 ;*************************************************************
;2315 =00
;2316 SELECT.LOWER:
;2317  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1008, 0818,0039,1980,0800,0000,1263  ;2318   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1009, 0000,003E,0180,0800,0000,0001  ;2319  RETURN1    ; Lower Controller Misconfigured
U 100A, 0810,0038,0180,0970,0000,100B  ;2320  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2321  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 100B, 0019,0016,01B0,09E8,0000,0002  ;2322   RETURN2   ; Let caller know that the controller
;2323       ; ...was configured properly
```

```
;2324  .PAGE
;2325  .TOC "   SELECT UPPER CONTROLLER"
;2326 ;*****************************************************************
;2327 ;++
;2328 ;
;2329 ; Functional Description:
;2330 ; ----------------------
;2331 ;
;2332 ; This subroutine can be called to select the upper
;2333 ; Controller on a MS780-E/H.
;2334 ; If the Upper controller is misconfigured then a
;2335 ; RETURN1 will be made.  Otherwise a RETURN2
;2336 ;
;2337 ; Calling Constraint: =00
;2338 ; ------------------
;2339 ;
;2340 ;
;2341 ; Inputs:
;2342 ; ------
;2343 ;
;2344 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2345 ;
;2346 ; Outputs:
;2347 ; -------
;2348 ;
;2349 ; RC[0D] will have the address of CNFG Register D
;2350 ;        ( 2000x010 )
;2351 ;
;2352 ; Side Effects:
;2353 ; ------------
;2354 ;
;2355 ; Contents of the following registers will lost:
;2356 ;
;2357 ; D
;2358 ;
;2359 ; The Upper controller on the MS780-E/H will be configured
;2360 ; when the subroutine is exitted with a RETURN2
;2361 ;--
;2362 ;*****************************************************************
;2363 =00
;2364 SELECT.UPPER:
;2365  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1020, 0818,0039,0980,0800,0000,1263  ;2366   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000,003E,0180,0800,0000,0001  ;2367  RETURN1     ; Upper Controller Misconfigured
U 1022, 0810,0038,0180,0970,0000,1023  ;2368  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2369  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1023, 0019,0016,8580,09E8,0000,0002  ;2370   RETURN2    ; Let caller know that the controller
;2371       ; ...was configured properly
;2372
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0974
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58                    Page   6

            ;2373  .PAGE
            ;2374  .TOC "   SBI Unjam Routine"
            ;2375 ;**
            ;2376 ; FUNCTIONAL DESCRIPTION:
            ;2377 ;
            ;2378 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2379 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2380 ; and finally HOLD for the last 16 cycles.
            ;2381 ;
            ;2382 ; CALLING CONSTRAINT:
            ;2383 ;
            ;2384 ; =0
            ;2385 ;
            ;2386 ; SIDE EFFECTS:
            ;2387 ;
            ;2388 ; D Reg destroyed
            ;2389 ;--
            ;2390    ;
            ;2391    ; assert hold for 16 cycles
            ;2392    ;
            ;2393 CLRSBI:
            ;2394 SBI.UNJAM:
            ;2395  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 1272, 0818,0038,6580,8000,0010,1054  ;2396      clk.ubcc  ; set micro cc's for next cycle
            ;2397 =0
            ;2398 SBI.UNJAM.1:
            ;2399  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2400       clk.ubcc,z?, ; ...
U 1054, 0819,0100,0580,8000,0010,1054  ;2401      j/sbi.unjam.1 ; test for completion
            ;2402    ;
            ;2403    ; assert hold and unjam for 16 cycles
            ;2404    ;
            ;2405  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 1055, 0818,0038,6580,8800,0010,1058  ;2406      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2407 =0
            ;2408 SBI.UNJAM.2:
            ;2409  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2410       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1058, 0819,0100,0580,8800,0010,1058  ;2411      z?,j/sbi.unjam.2 ; ...
            ;2412    ;
            ;2413    ; assert hold for 16 cycles
            ;2414    ;
            ;2415  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1059, 0818,0038,6580,8000,0010,105C  ;2416      clk.ubcc  ; and set cc's for next cycle
            ;2417 =0
            ;2418 SBI.UNJAM.3:
            ;2419  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2420       clk.ubcc,z?, ; ...
U 105C, 0819,0100,0580,8000,0010,105C  ;2421      j/sbi.unjam.3 ; ...
U 105D, 0000,003E,0180,0800,0000,0001  ;2422  returnl   ; exit
            ;2423
```

```
            ;2424  .PAGE
            ;2425  .TOC "   RESET SUBTEST NUMBER"
            ;2426 ;**
            ;2427 ; FUNCTIONAL DESCRIPTION:
            ;2428 ;
            ;2429 ; Since some  of the tests will have to be restarted when two
            ;2430 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2431 ; the subtest counter to zero ( only for thoes tests that have
            ;2432 ; more than one subtest). This subroutine may also be necessary
            ;2433 ; when a test is being loop on.
            ;2434 ;
            ;2435 ; CALLING CONSTRAINT:
            ;2436 ;
            ;2437 ; =0
            ;2438 ; SIDE EFFECTS:
            ;2439 ;
            ;2440 ; D is destroyed
            ;2441 ;
            ;2442 ;--
            ;2443 RESET.SUBTST:
            ;2444  RC[0F]_0,    ; Reset subtest number
U 1273, 0003,003E,0180,09F8,0000,0001  ;2445   RETURN1    ; ...and return
```

```
            ;2446  .PAGE
            ;2447  .TOC "   Initialize the SBI"
            ;2448 ;++
            ;2449 ; FUNCTIONAL DESCRIPTION:
            ;2450 ;
            ;2451 ; This routine performs the following functions:
            ;2452 ;
            ;2453 ;  Executes an SBI Unjam
            ;2454 ;  Clears the SBI Fault Latch
            ;2455 ;  Clears the SBI Error Register
            ;2456 ;
            ;2457 ; CALLING CONSTRAINT:
            ;2458 ;
            ;2459 ; =0
            ;2460 ;
            ;2461 ; SIDE EFFECTS:
            ;2462 ;
            ;2463 ; D & Q Register destroyed
            ;2464 ;--
            ;2465 =0
            ;2466 SBI.INIT:
U 107C, 0000.003D,0180.0800.0000.1272  ;2467  call[sbi.unjam]  ; Unjam the SBI
U 107D, 0818.0038,C180.0800.0000.1274  ;2468  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1274, 0801.0028,6580.3C00.0000.1275  ;2469  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1275, 0F00.003C,6D80.3C00.0000.1276  ;2470  id[fault]_d,d_0
U 1276, 0000.003C,6D80.3C00.0000.1277  ;2471  id[fault]_d
U 1277, 0000.003E,6580.3C00.0000.0001  ;2472  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2473
```

```
            ;2474  .PAGE
            ;2475  .TOC "   Enable Cache"
            ;2476 ;++
            ;2477 ; FUNCTIONAL DESCRIPTION:
            ;2478 ;
            ;2479 ; This routine enables cache group 0 by clearing the force
            ;2480 ; miss bit in the maintenance register.
            ;2481 ;
            ;2482 ; CALLING CONSTRAINT:
            ;2483 ;
            ;2484 ; =0
            ;2485 ;
            ;2486 ; SIDE EFFECTS:
            ;2487 ;
            ;2488 ; D, Q AND SC Registers destroyed
            ;2489 ;--
            ;2490 ENABLE.CACHE:
U 1278, 0000,003C,75F0,2C00,0000,1279  ;2491  q_id[maint]  ; Get current maintenance register
U 1279, 0000,003C,6580,0800,0084,727A  ;2492  sc_k[.10]  ; Setup to clear bit 16
U 127A, 0801,2034,0180,0800,0000,127B  ;2493  d_q.and.mask  ; Clear bit 16
U 127B, 0000,003E,7580,3C00,0000,0001  ;2494  id[maint]_d,return1 ; Rewrite register and return
            ;2495
```

```
            ;2496  .PAGE
            ;2497  .TOC "   Disable Cache Routine"
            ;2498 ;++
            ;2499 ; FUNCTIONAL DESCRIPTION:
            ;2500 ;
            ;2501 ; This routine disables cache hits by setting bits <16:15>
            ;2502 ; in the maintenance register.
            ;2503 ;
            ;2504 ; CALLING SEQUENCE:
            ;2505 ;
            ;2506 ; =0
            ;2507 ;
            ;2508 ; SIDE EFFECTS:
            ;2509 ;
            ;2510 ; D & Q Registers destroyed
            ;2511 ;--
            ;2512 DISABLE.CACHE:
U 127C. 0818.0038.4580.0800.0000.127D  ;2513  d_k[.8000]  ; ... and seed constant
U 127D. 0500.003C.0180.0800.0000.127E  ;2514  d_d.left  ; Generate final constant
U 127E. 0819.0030.4580.0800.0000.127F  ;2515  d_d.or.k[.8000]  ; ... = 00018000
U 127F. 0000.003E.7580.3C00.0000.0001  ;2516  id[maint]_d,return1 ; Disable cache and return
            ;2517
```

```
            ;2518  .PAGE
            ;2519  .TOC "   Check for Interrupt Routine"
            ;2520 ;++
            ;2521 ; FUNCTIONAL DESCRIPTION:
            ;2522 ;
            ;2523 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2524 ; If an interrupt is active, the following registers are setup:
            ;2525 ;  RC[8] - Gets the VECTOR register
            ;2526 ;  RC[7] - Gets the SBI ERROR register
            ;2527 ;  RC[6] - Gets the FAULT register
            ;2528 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2529 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2530 ;
            ;2531 ; CALLING CONSTRAINT:
            ;2532 ;
            ;2533 ; =00
            ;2534 ;
            ;2535 ; SIDE EFFECTS:
            ;2536 ;
            ;2537 ; D & Q are destroyed
            ;2538 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2539 ;--
            ;2540 CHECK_INTERRUPT:
            ;2541   ;
            ;2542   ; First, enable the fault and RDS/CRD interrupts
            ;2543   ;
U 1280, 0818,0038,4580,0800,0000,1281  ;2544  d_k[.8000]  ; Get data to set interrupt enable
            ;2545  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1281, 0500,003C,6580,3C00,0000,1282  ;2546      d_d.left
U 1282, 0100,003C,0180,0800,0000,1283  ;2547  d_d.left2  ; D = 40000
U 1283, 0000,003C,6D80,3C00,0000,1284  ;2548  id[fault]_d  ; Set fault interrupt enable
            ;2549  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1284, 0F03,003C,0180,09A8,4000,1285  ;2550      rc[5]_0  ; and clear interrupt occurred flag
U 1285, 0000,003C,3D80,3C00,0000,1286  ;2551  id[psl]_d  ; Clear the PSL
U 1286, 0000,003C,6580,3C00,0000,1287  ;2552  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1287, 0000,003C,6D80,3C00,0000,1288  ;2553  id[fault]_d  ; Clear FAULT Interrupt Enable
U 1288, 0000,0E3C,0180,0800,0000,1004  ;2554  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2555 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2556  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2557  return1   ; No interrupt
            ;2558 =111
            ;2559 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,1289  ;2560  q_id[vector]  ; Get ID Vector Register
U 1289, 0001,203C,65F0,2DC0,0000,128A  ;2561  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 128A, 0001,203C,6DF0,2DB8,0000,128B  ;2562  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 128B, 0001,203C,0180,09B0,0000,128C  ;2563  rc[6]_q   ; Save falut reg
U 128C, 0018,003A,0580,09A8,0000,0002  ;2564  rc[5]_k[.1],return2 ; Set error flag and return
            ;2565
```

```
            ;2566  .PAGE
            ;2567  .TOC "    CHECK UTRAP"
            ;2568 ;++
            ;2569 ;FUNCTIONAL DESCRIPTION:
            ;2570 ;
            ;2571 ; This subroutine is used to check wether a Utrap occured.
            ;2572 ; It takes the contents of the Save Utrap flags register
            ;2573 ; R[0E], and compares them to the contrnts of the Utrap
            ;2574 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2575 ; according to the results of the test  (i.e., contents of
            ;2576 ; these registers are equal or not).
            ;2577 ; CALLING CONSTRAINT:
            ;2578 ;
            ;2579 ; =00
            ;2580 ;
            ;2581 ; INPUTS:
            ;2582 ;
            ;2583 ; R[0E]- Saved Utrap Mask
            ;2584 ; R[0F]- Expected Utrap Mask
            ;2585 ;
            ;2586 ;--
            ;2587 CHECK_UTRAP:
U 128D, 0800,003C,0180,0A70,0000,128E  ;2588  D_R[0E]    ; Reg D is loaded with the trap mask
U 128E, 0001,003C,0180,09C0,0000,128F  ;2589  RC[08]_D   ; Save expected Utrap flag for error logout
U 128F, 0800,003C,0180,0A78,0000,1290  ;2590  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1290, 0001,003C,0180,09B8,0000,1291  ;2591  RC[07]_D   ; Save in RC[07]
            ;2592  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1291, 000D,0020,0180,0A70,0010,1292  ;2593         CLK.UBCC
U 1292, 0003,013C,0180,0AF8,0000,108C  ;2594  Z?,R[0F]_0   ; Branch if no traps
U 108C, 0000,003E,0180,0800,0000,0002  ;2595 =0 RETURN2    ; Utrap occured
U 108D, 0000,003E,0180,0800,0000,0001  ;2596  RETURN1    ; No Utrap  return
            ;2597 ;
```

```
            ;2598   .PAGE
            ;2599   .TOC "    Set Trap Mask"
            ;2600 ;**
            ;2601 ; FUNCTIONAL DESCRIPTION:
            ;2602 ;
            ;2603 ; This routine is used to set a bit in the Micro Trap Mask
            ;2604 ; R[0F].
            ;2605 ;
            ;2606 ; CALLING CONSTRAINT:
            ;2607 ;
            ;2608 ; =0
            ;2609 ;
            ;2610 ; INPUTS:
            ;2611 ;
            ;2612 ; SC - Contains bit number to set.
            ;2613 ;
            ;2614 ; OUTPUTS:
            ;2615 ;
            ;2616 ; rc[0E] - Contains the current trap mask
            ;2617 ;
            ;2618 ; SIDE EFFECTS:
            ;2619 ;
            ;2620 ; d regsiter is destroyed
            ;2621 ;--
            ;2622 SET_TRAP_MASK:
U 1293, 0800,003C,0180,0A78,0000,1294  ;2623  d_r[0f]
U 1294, 0801,001C,0180,0AF8,0000,1295  ;2624  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1295, 0001,003E,0180,0AF0,0000,0001  ;2625  r[0E]_d,return1    ; Save mask and return
            ;2626
```

```
            ;2627  .PAGE
            ;2628  .TOC "   FIND MS780-E MEMORY"
            ;2629 ;++
            ;2630 ; FUNCTIONAL DESCRIPTION:
            ;2631 ;
            ;2632 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2633 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2634 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2635 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2636 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2637 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2638 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2639 ;     (zero).
            ;2640 ;     All the other MS780-E/H's found will have their starting  address
            ;2641 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2642 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2643 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2644 ;     current MS780-E unit under test.
            ;2645 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2646 ;     set to maximum.
            ;2647 ;
            ;2648 ; CALLING CONSTRAINT:
            ;2649 ;
            ;2650 ; =00
            ;2651 ;
            ;2652 ; OUTPUTS:
            ;2653 ;
            ;2654 ; rc[0e] - Contains address of Configuration Register
            ;2655 ; r[0]  - Contains TR level
            ;2656 ;
            ;2657 ; SIDE EFFECTS:
            ;2658 ;
            ;2659 ; D register is destroyed.
            ;2660 ;--
            ;2661 =0
            ;2662 FIND.MS780E:
U 1094. 0000,003D,0180,0800,0000,107C  ;2663  call[sbi.init]  ; Make sure SBI is enabled
U 1095. 0003,003C,0180,0988,0000,1296  ;2664  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1296. 0818,0038,1980,0800,0000,1297  ;2665  d_k[zero]  ; Clear MS780-E/H counter
U 1297. 0000,003C,F980,3C00,0000,1298  ;2666  id[3e]_d  ; ...
U 1298. 0018,0038,0580,0A80,0000,1299  ;2667  r[0]_k[.1]  ; Init TR counter
U 1299. 0F03,003C,0180,09F0,0000,109C  ;2668  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2669      ; ...CNFG A Reg
            ;2670 =0
            ;2671 FIND.MEM.LOOP:
U 109C. 0800,003D,0180,0A00,0000,12C0  ;2672  d_r[0],call[generate.io]; Generate address of TR level
U 109D. 0001,003C,0180,09F0,0200,110A  ;2673  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 110A. 0000,003D,8980,0800,0084,7293  ;2674 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2675  d[long]_cache.p, ; Read the specified tr level
U 110B. 0000,003C,0180,C970,0000,129A  ;2676    ic_rc[0e]  ; ...
U 129A. 0000,003C,0180,0A78,0010,129B  ;2677  alu_r[0f],clk.ubcc ; Check for no response
U 129B. 0001,013C,0180,0880,0082,113C  ;2678  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 113C. 0000,003C,0180,0800,0000,129C  ;2679 =0 j/check.adpt.code ; Check for a memory
U 113D. 0000,003C,0180,0800,0000,12BC  ;2680  j/find.mem.lp1  ; Try next tr
            ;2681 CHECK.ADPT.CODE:
```

```
U 129C, 0000,003C,1D80,0800,1404,729D  ;2682  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 129D, 0000,163C,0180,0800,0000,1018  ;2683  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,729E  ;2684 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,729F  ;2685  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,12BC  ;2686  j/find.mem.lp1  ; Not a memory
U 101B, 0000,003C,0180,0800,0000,12BC  ;2687  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,72A4  ;2688  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,12BC  ;2689  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,12A8  ;2690  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,12A8  ;2691  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2692   ;
            ;2693   ; Found an MS780-A or -C. Set the starting address
            ;2694   ; to maximum.
            ;2695   ;
U 129E, 0818,0038,5D80,0800,0000,12A0  ;2696 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 129F, 0818,0038,0580,0800,0000,12A0  ;2697 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2698 MS780.COMMON:
U 12A0, 0819,0030,7180,0800,0000,12A1  ;2699  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 12A1, 0000,003C,01F8,0803,0080,D2A2  ;2700  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 12A2, 0D00,003C,0180,0800,0000,12A3  ;2701  d_dal.sc  ; Put data in bits<29:14>
            ;2702  cache.p_d[long], ; Set the starting address to maximum
U 12A3, 0000,003C,0180,A800,0000,12BC  ;2703       j/find.mem.lp1 ; and continue TR scan
            ;2704   ;
            ;2705   ; Found an MA780. Set the starting address to maximum
            ;2706   ;
            ;2707 MA780.MAX:
U 12A4, 0818,0038,39F8,0803,0000,12A5  ;2708  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 12A5, 0D00,003C,0180,0803,0000,12A6  ;2709  d_dal.sc,va_va+4 ; Put in proper position
U 12A6, 0000,003C,0180,0803,0000,12A7  ;2710  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2711  cache.p_d[long], ; Set starting address to maximum
U 12A7, 0000,003C,0180,A800,0000,12BC  ;2712       j/find.mem.lp1
            ;2713 ;
            ;2714 ; Found an MS780E
            ;2715 ;
            ;2716 FOUND.MS780E:
U 12A8, 0000,003C,F9F0,2C00,0000,12A9  ;2717  q_id[3e]  ; Set up to see how many MS780-E's
            ;2718  alu_q.xor.k[.3], ; Are there Maximum units?
U 12A9, 0019,2020,0D80,0800,0010,12AB  ;2719   clk.ubcc ; Check condition codes
U 12AB, 0000,013C,0180,0800,0000,113E  ;2720  z?   ; Are we at maximum units for system
U 113E, 0000,003C,0180,0800,0000,12AC  ;2721 =0 J/ms780e.tst  ; No go find how many units ther are
U 113F, 0818,0038,C180,0800,0000,1140  ;2722  d_k[.ffff]  ; Yes set address to maximum
U 1140, 0000,003D,0180,0800,0000,12C4  ;2723 =0 call[set.start.addr] ; .....
U 1141, 0000,003C,0180,0800,0000,12BC  ;2724  j/find.mem.lp1  ; Go check to see if we are done
            ;2725 ;
            ;2726 MS780E.TST:
            ;2727  alu_rc[01],  ; Check "Found MS780E Flag"
U 12AC, 0010,0038,0180,0908,0010,12AD  ;2728   clk.ubcc ; Check condition codes
U 12AD, 0810,0138,0180,0970,0000,1142  ;2729  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2730 =0 j/not.first.ms780e, ; No
U 1142, 0818,0038,C180,0800,0000,1146  ;2731   d_k[.ffff] ; Set starting address
U 1143, 0001,003C,0180,0988,0000,12AE  ;2732  rc[01]_d  ; Yes put base address in rc[01]
U 12AE, 0818,0038,7980,0800,0000,12AF  ;2733  d_k[.30]  ; Set up SC to point to first unit
U 12AF, 0019,0014,F580,0800,0082,12B0  ;2734  alu_d+k[.a],sc_alu ; ...
U 12B0, 0810,0038,0180,0908,0000,12B1  ;2735  d_rc[01]  ; Put base address of unit in D
U 12B1, 0000,003C,0180,3400,0000,12B2  ;2736  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 12B2, 0F00,003C,0180,0800,0000,1144  ;2737  d_0   ; Prepare to set starting address to Zero
U 1144, 0000,003D,0180,0800,0000,12C4  ;2738 =0 call[set.start.addr] ; Go do it
          ;2739  j/find.mem.lp1,  ; Check to see if we are done
U 1145, 0003,003C,0180,09E8,0000,12BC  ;2740   rc[0d]_0 ; ....
          ;2741 =0
          ;2742 NOT.FIRST.MS780E:
U 1146, 0000,003D,0180,0800,0000,12C4  ;2743  call[set.start.addr] ; Go set starting address to maximum
U 1147, 0000,003C,F9F0,2C00,0000,12B3  ;2744  q_id[3e]  ; Get the number of MS780-Es found
U 12B3, 0819,2014,0580,0800,0000,12B4  ;2745  d_q+k[.1]  ; Increment this counter
U 12B4, 0000,003C,F980,3C00,0000,12B5  ;2746  id[3e]_d  ; Save the counter
U 12B5, 0018,0038,11C0,0800,0000,12B6  ;2747  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2748      ; ...were the I/O base addresses will be stored
U 12B6, 081D,2000,7980,0800,0000,12B7  ;2749  d_q-d,k[.30]  ; ... adjust pointer
U 12B7, 0819,0014,7980,0800,0000,12B8  ;2750  d_d+k[.30]  ; Point the SC to the ID register
U 12B8, 0000,003C,F580,0800,0000,12B9  ;2751  k[.a]   ; ...to receive I/O base address
U 12B9, 0019,0014,F580,0800,0082,12BA  ;2752  alu_d+k[.a],sc_alu ; ...
U 12BA, 0810,0038,0180,0970,0000,12BB  ;2753  d_rc[0e]  ; Get base address to d
U 12BB, 0000,003C,0180,3400,0000,12BC  ;2754  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2755   ;
          ;2756   ; Try next tr
          ;2757   ;
          ;2758 FIND.MEM.LP1:
U 12BC, 0818,0014,0580,0A80,0000,12BD  ;2759  r[0]_la+k[.1],d_alu ; Increment TR level
          ;2760  alu_d.and.k[.f],
U 12BD, 0019,0034,6180,0800,0010,12BE  ;2761      clk.ubcc  ; Check if all done
U 12BE, 0000,013C,0180,0800,0000,1148  ;2762  z?   ; Branch if yes
U 1148, 0000,003C,0180,0800,0000,109C  ;2763 =0 j/find.mem.loop  ; Try next TR
U 1149, 0810,0038,0180,0908,0010,12BF  ;2764  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 12BF, 0000,013C,0180,0800,0000,114A  ;2765  z?   ; Check condition codes
U 114A, 0001,003E,0180,09F0,0000,0002  ;2766 =0 return2,rc[0e]_d ; Yes MS780E was found
U 114B, 0000,003E,0180,0800,0000,0001  ;2767  return1   ; No MS780E found
          ;2768
```

```
            ;2769   .PAGE
            ;2770   .TOC "   Generate IO Address"
            ;2771 ;++
            ;2772 ; FUNCTIONAL DESCRIPTION:
            ;2773 ;
            ;2774 ; This routine is used to generate an SBI Configuration Register
            ;2775 ; address from a TR level.
            ;2776 ;
            ;2777 ; CALLING CONSTRAINT:
            ;2778 ;
            ;2779 ; =0
            ;2780 ;
            ;2781 ; INPUTS:
            ;2782 ;
            ;2783 ; D - Contains TR level
            ;2784 ;
            ;2785 ; OUTPUTS:
            ;2786 ;
            ;2787 ; D - Contains SBI IO address
            ;2788 ;--
            ;2789 GENERATE.IO:
U 12C0, 0000.003C.89F8.0800.0084.72C1  ;2790  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 12C1, 0D00.003C.8D80.0800.0084.72C2  ;2791  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 12C2, 0000.003C.0980.0800.0084.B2C3  ;2792  sc_sc-k[.2]  ; insert bit 29
U 12C3, 0801.001E.0180.0800.0000.0001  ;2793  d_d.ornot.mask,return1 ; and return
            ;2794
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0986
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58                    Page   8

            ;2795  .PAGE
            ;2796  .TOC "   Set Starting Address"
            ;2797 ;++
            ;2798 ; FUNCTIONAL DESCRIPTION:
            ;2799 ;
            ;2800 ; This routine is used to set the starting address of the
            ;2801 ; memory to the specified value.
            ;2802 ;
            ;2803 ; CALLING CONSTRAINT:
            ;2804 ;
            ;2805 ; =0
            ;2806 ;
            ;2807 ; INPUTS:
            ;2808 ;
            ;2809 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2810 ;     (B Register Image divided by 2**16)
            ;2811 ; rc[0e] - Contains Address of Register A
            ;2812 ;
            ;2813 ; OUTPUTS:
            ;2814 ;
            ;2815 ; d - Contains aligned starting address (B Register Image)
            ;2816 ;
            ;2817 ; SIDE EFFECTS:
            ;2818 ;
            ;2819 ; d,q,va, and sc are destroyed
            ;2820 ;--
            ;2821 SET.START.ADDR:
U 12C4, 0010,0038,6580,0970,0284,72C5  ;2822  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 12C5, 0000,003C,01F8,0803,0000,12C6  ;2823  va_va+4,q_0  ; Set address to Register B
            ;2824  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 12C6, 0D00,003C,0980,0800,0084,B2C7  ;2825       sc_sc-k[.2]  ; Setup to insert bit 14
U 12C7, 0801,001C,0180,0800,0000,12C8  ;2826  d_d.ornot.mask  ; Insert Write Enable bit
U 12C8, 0000,003E,0180,A800,0000,0001  ;2827  cache.p_d[long],return1 ; Set the starting address and return
            ;2828
```

```
            ;2829  .PAGE
            ;2830  .TOC "   ENABLE NEXT MS780-E"
            ;2831 ;++
            ;2832 ; FUNCTIONAL DESCRIPTION:
            ;2833 ;
            ;2834 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2835 ;    They will be tested  separately, according to the TR level at which
            ;2836 ;    they are located,  starting with the one  at the lowest level first.
            ;2837 ;          When  a test has finished  with the first unit, it will have to call
            ;2838 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2839 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2840 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2841 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2842 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2843 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2844 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2845 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2846 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2847 ;
            ;2848 ; CALLING CONSTRAINT:
            ;2849 ;
            ;2850 ;  =00
            ;2851 ;
            ;2852 ;--
            ;2853 ENABLE.NEXT.MS780E:
U 12C9, 0000,003C,F9F0,2C00,0000,12CA  ;2854  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2855  ALU_Q,   ; Check to see if it is zero
U 12CA, 0001,203C,0180,0800,0010,12CB  ;2856      CLK.UBCC  ; Check Condition Codes
U 12CB, 0000,013C,0180,0800,0000,114C  ;2857  Z?   ; ...for zero
U 114C, 0000,003C,0180,0800,0000,12CC  ;2858 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 114D, 0000,003E,0180,0800,0000,0002  ;2859  RETURN2   ; Zero No more units to test
            ;2860 ENABLE.NEXT:
U 12CC, 0818,0038,C180,0800,0000,114E  ;2861  D_K[.FFFF]  ; Load D with Maximum Starting address
U 114E, 0000,003D,0180,0800,0000,12C4  ;2862 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 114F, 0818,0038,7980,0800,0000,12CD  ;2863  D_K[.30]  ; Prepare to point to the ID register holding
            ;2864     ; ...the I/O Base address of the next MS780-E
U 12CD, 0819,0014,1180,0800,0000,12CE  ;2865  D_D+K[.4]  ; ...
U 12CE, 0000,003C,F580,0800,0000,12CF  ;2866  K[.A]   ; ...
U 12CF, 0819,0014,F580,0800,0000,12D0  ;2867  D_D+K[.A]  ; ...
U 12D0, 0000,003C,F9F0,2C00,0000,12D1  ;2868  Q_ID[3E]  ; Get MS780-E Counter
            ;2869  D_D-Q,   ; D now holds the pointer to the ID register
U 12D1, 081D,0000,0180,0800,0082,12D2  ;2870    SC_ALU  ; ...
U 12D2, 0000,003C,01F0,2400,0000,12D3  ;2871  Q_ID(SC)  ; Q gets adrress of next MS780E
U 12D3, 0001,203C,0180,09F0,0000,12D4  ;2872  RC[0E]_Q  ; Save it also in RC[0E]
U 12D4, 0F03,003C,0180,09E8,0000,1150  ;2873  RC[0D]_0,D_0  ; Set starting address to zero
U 1150, 0000,003D,0180,0800,0000,12C4  ;2874 =0 CALL[SET.START.ADDR] ; ....
U 1151, 0F00,003C,F9F0,2C00,0000,12D5  ;2875  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 12D5, 081D,2008,0180,0800,0000,12D6  ;2876  D_Q-D-1   ; Subtract 1 from total
U 12D6, 0000,003C,F980,3C00,0000,1152  ;2877  ID[3E]_D  ; Adjust the pointer
U 1152, 0000,003D,0180,0800,0000,1273  ;2878 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 1153, 0000,003E,0180,0800,0000,0001  ;2879  RETURN1   ; Tell caller another unit was found
            ;2880
```

```
                ;2881   .PAGE
                ;2882   .TOC "    GET MEMORY SIZE"
                ;2883 ;*************************************************************
                ;2884 ;++
                ;2885 ;
                ;2886 ; Functional Description:
                ;2887 ; ----------------------
                ;2888 ;
                ;2889 ; This subroutine can be used to get the size of the memory.
                ;2890 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
                ;2891 ; increments.
                ;2892 ;
                ;2893 ; Calling Constraint: =0
                ;2894 ; ------------------
                ;2895 ;
                ;2896 ;
                ;2897 ; Inputs:
                ;2898 ; ------
                ;2899 ;
                ;2900 ; RC[0E] Must have the Address of CNFG Register A
                ;2901 ;
                ;2902 ; Outputs:
                ;2903 ; -------
                ;2904 ;
                ;2905 ; Register D will contain upon exit the size of the
                ;2906 ; memory aligned on Mega-byte boundary.
                ;2907 ;
                ;2908 ; Side Effects:
                ;2909 ; ------------
                ;2910 ;
                ;2911 ; The contents of the following registers will be lost:
                ;2912 ;
                ;2913 ; D, SC, Q
                ;2914 ;
                ;2915 ;--
                ;2916 ;*************************************************************
                ;2917 GET.MEMORY.SIZE:
U 12D7, 0010,0038,0180,0970,0200,12D8   ;2918  VA_RC[0E]    ; Point to CNFG Reg A
U 12D8, 0000,003C,0180,C800,0000,12D9   ;2919  D[LONG]_CACHE.P    ; Read the register
U 12D9, 001B,0000,D980,0800,0082,12DA   ;2920  ALU_0-K[.9],SC_ALU   ; Get -10 in the SC
U 12DA, 0D00,003C,0180,0800,0000,12DB   ;2921  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 12DB, 0819,0034,5580,0800,0000,1024   ;2922  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
                ;2923 =00 CALL[64K.OR.256K],   ; Find out the size of the arrays
U 1024, 0000,003D,01E0,0800,0000,126D   ;2924   Q_D    ; Save Memory size bits in Q
U 1025, 0000,003D,0D80,0800,0084,703C   ;2925  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
                ;2926       ; ...arrays on the controller
                ;2927 ;
                ;2928 ; The duplication of the next two return status is necessary so the returning
                ;2929 ; subroutine (64k.or.256k) can be used in other modules without any
                ;2930 ; modifications.
                ;2931 ;
U 1026, 0819,2014,0580,0800,0000,1027   ;2932  D_Q+K[.1]    ; RET2 Increment memory size by one
                ;2933       ; ...(found 1 Meg arrays)
U 1027, 0819,2014,0580,0800,0000,12DC   ;2934  D_Q+K[.1]    ; RET3 Increment memory size by 1
                ;2935       ; ...(found 4 Meg arrays)
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0989
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58                    Page   8

U 12DC. 0000,003C,21F8,0800,0084,72DD  ;2936  SC_K[.14],Q_0    ; Prepare to shift to Megabyte position
          ;2937  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 12DD. 0D00,003E,0180,0800,0000,0001  ;2938   RETURN1    ; Return with memory size in Register D
          ;2939
```

```
          ;2940  .PAGE
          ;2941  .TOC "   ERDAT ROUTINE"
          ;2942 ;
          ;2943 ;////////////////////////////////////////////////////////////////////////////
          ;2944 ;++
          ;2945 ;       THIS SUBROUTINE GETS THE ERROR DATA FROM ID[25-2X]
          ;2946 ; AND PUTS THAT DATA IN RC[0E]-RC[0X].
          ;2947 ;
          ;2948 ; (X DEPENDS ON HOW MANY ERRORS OCCURRED; MAXIMUM OF 6 DECIMAL)
          ;2949 ;
          ;2950 ; IT IS ASSUMMED THAT THE NUMBER OF ERRORS IS CONTAINED
          ;2951 ; IN THE STATE REGISTER.
          ;2952 ; (ALL F'S IS THE FLAG FOR END OF ERROR DATA).
          ;2953 ;--
          ;2954 ;////////////////////////////////////////////////////////////////////////////
          ;2955 ;
U 12DE, 0000,003C,8580,0800,0184,72DF  ;2956 ERDAT: FE&SC_K[.C]   ; INIT RC ADRS TO C
U 12DF, 0000,003C,B980,0800,0084,92E0  ;2957  SC_SC+K[.19]   ; INIT ID ADRS TO 25 (HEX)
U 12E0, 0000,003C,01F0,2400,0000,12E1  ;2958 ERDLP: Q_ID(SC)
U 12E1, 0000,003C,0180,0800,0181,D2E2  ;2959  FE_SC+1,SC_FE   ; INCR ID ADRS,SC_RC ADRS
U 12E2, 0001,203C,0180,0838,0000,12E3  ;2960  RC(SC)_ALU,ALU_Q
U 12E3, 0000,003C,0580,0800,0185,B2E4  ;2961  SC_FE,FE_SC-K[.1]  ; SC_ID ADRS,FE_DECR'D RC ADRS
U 12E4, 0000,003C,0580,0800,1414,B2E5  ;2962  STATE_STATE-K[.1],CLK.UBCC ; DECR ERROR COUNTER
U 12E5, 0000,123C,0180,0800,0000,102B  ;2963  EALU.Z?    ; DONE LOOP ERROR # TIMES?
U 102B, 0000,003C,0180,0800,0000,12E0  ;2964 =1011 J/ERDLP    ; NOPE
U 102F, 0000,003C,0180,0800,0081,12E6  ;2965  SC_FE    ; PUT END FLAG IN RC[X]
          ;2966  RETURN1,   ; ...AND RETURN TO ERROR CALL.
U 12E6, 001B,0002,0580,0838,0000,0001  ;2967   RC(SC)_ALU,ALU_0-K[.1]
          ;2968
```

```
            ;2969  .PAGE
            ;2970  .TOC "   Set Diagnostic Mode Routine"
            ;2971 ;++
            ;2972 ; FUNCTIONAL DESCRIPTION:
            ;2973 ;
            ;2974 ; This routine is used to set the specified diagnostic mode
            ;2975 ; in Register B. Other bits in the register remain unchanged.
            ;2976 ;
            ;2977 ; CALLING CONSTRAINT:
            ;2978 ;
            ;2979 ; =0
            ;2980 ;
            ;2981 ; INPUTS:
            ;2982 ;
            ;2983 ; d - Contains the mode to set in bits<1:0>
            ;2984 ; rc[0e] - Contains base address of controller
            ;2985 ;
            ;2986 ; SIDE EFFECTS:
            ;2987 ;
            ;2988 ; d,va,sc are destroyed
            ;2989 ;--
            ;2990 SET_DIAG_MODE:
U 12E7, 0010,0038,D9F8,0970,0284,72E8  ;2991  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 12E8, 0D00,003C,0180,0803,0000,12E9  ;2992  va_va*4,d_dal.sc ; Shift specified mode into position
U 12E9, 0000,003C,01E0,0800,0000,12EA  ;2993  q_d   ; Save the diagnostic mode bits
U 12EA, 0000,003C,0180,C800,0000,12EB  ;2994  d[long]_cache.p  ; Read the contents of the CNFG register B
U 12EB, 081D,0030,0180,0800,0000,12EC  ;2995  d_d.or.q  ; Set the diagnostic mode bits
U 12EC, 0000,003E,0180,A800,0000,0001  ;2996  cache.p_d[long],return1 ; Set the specified mode and return
            ;2997
```

```
;2998  .PAGE
;2999  .TOC "   Check For Odd or Even TR"
;3000 ;*************************************************************
;3001 ;**
;3002 ;
;3003 ; Functional Description:
;3004 ; ---------------------
;3005 ;
;3006 ;
;3007 ;      This subroutine will  determine  base  on  the  contents  of
;3008 ; RC[0E]  (which  contains  the  I/O  base register address for the
;3009 ; MS780-E/H currently under test), whether the memory is at Even or
;3010 ; Odd TR level.
;3011 ;
;3012 ;
;3013 ; Calling Constraint: =00
;3014 ; ------------------
;3015 ;
;3016 ;
;3017 ; Inputs:
;3018 ; ------
;3019 ;
;3020 ; RC[0E] must hold the I/O base address of the memory under test
;3021 ;
;3022 ;
;3023 ; Outputs:
;3024 ; -------
;3025 ;
;3026 ; There are no specific outputs for this routine.
;3027 ; The only output will be the RETURN mode:
;3028 ;
;3029 ;  RETURN1 for EVEN TR
;3030 ;
;3031 ;  RETURN2 for ODD TR
;3032 ;
;3033 ;
;3034 ; Side Effects:
;3035 ; ------------
;3036 ;
;3037 ; The contents of the following registers will be lost
;3038 ;
;3039 ; D, SC
;3040 ;
;3041 ;--
;3042 ;*************************************************************
;3043 ODD.EVEN.TR:
U 12ED, 081B,001C,B580,0800,0000,12EE  ;3044  D_NOT.K[.C]    ; Load D register with negative x12
U 12EE, 0001,003C,0180,0800,0082,12EF  ;3045  SC_D     ; Prepare SC for right shift of 12
U 12EF, 0810,0038,0180,0970,0000,12F0  ;3046  D_RC[0E]    ; Get the I/O base address
U 12F0, 0000,003C,01F8,0800,0000,12F1  ;3047  Q_0     ; Clear Q Reg
U 12F1, 0D00,003C,0180,0800,0000,12F2  ;3048  D_DAL.SC    ; Shift D right 13 bits
U 12F2, 0000,193C,0180,0800,0000,103E  ;3049  D[-0?     ; Is bit 0 of D set?
U 103E, 0000,003E,0180,0800,0000,0001  ;3050 =1110 RETURN1     ; EVEN TR
U 103F, 0000,003E,0180,0800,0000,0002  ;3051  RETURN2     ; ODD TR
          ;3052
```

```
;3053
;3054
;3055
;3056
```

```
;3057  .PAGE
;3058  .TOC "   REED MULLER ENCODING ROUTINE"
;3059 ;*************************************************************
;3060 ;**
;3061 ;
;3062 ; Functional Description:
;3063 ; ----------------------
;3064 ;
;3065 ;      This subroutine is used to apply the Reed-Muller
;3066 ; encoding  algorithm to a 16 bit  vector with an error
;3067 ; handling capability of up to 3 bits per 32 bit vector.
;3068 ;
;3069 ;
;3070 ; NOTE: The DESIGN SPEC for the MS780-E/H diagnostic
;3071 ;  includes a theoretical discussion of the
;3072 ;  Reed-Muller codes as well as the algorithm
;3073 ;  for the following subroutines:
;3074 ;
;3075 ;  ENCODE
;3076 ;
;3077 ;  DECODE
;3078 ;
;3079 ;  FIRST.ORDER
;3080 ;
;3081 ;  SECOND.ORDER
;3082 ;
;3083 ;
;3084 ; Calling Constraint: =0
;3085 ; ------------------
;3086 ;
;3087 ;
;3088 ; Inputs:
;3089 ; ------
;3090 ;
;3091 ; RC[4] must contain the 16 bit vector that should
;3092 ; be encoded.
;3093 ;
;3094 ; Outputs:
;3095 ; -------
;3096 ;
;3097 ; RC[1] will contain upon exit the encoded vector
;3098 ;
;3099 ;
;3100 ; Side Effects:
;3101 ; ------------
;3102 ;
;3103 ; The contents of the following registers will be lost:
;3104 ;
;3105 ; R[0] through R[0B], RC[1], T1 through T6
;3106 ;
;3107 ;--
;3108 ;*************************************************************
;3109 ENCODE:
U 12F3, 0018,0038,0580,0AD0,0000,12F4  ;3110  R[0A]_K[.1]   ; Initialize SYMBOL_POINTER (to the
;3111       ; ...bits of the INFO_SYMBOLS vector)
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 0995
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58                    Page   8

U 12F4. 0003,003C,0180,0988,0000,12F5  ;3112  RC[1]_0     ; Clear CODE_VECTOR location
U 12F5. 0810,0038,0180,0920,0000,12F6  ;3113  D_RC[4]     ; Get INFO_SYMBOLS to be coded
U 12F6. 0001,003C,0180,0990,0000,12F7  ;3114  RC[2]_D     ; Initialize INFO_SYMBOLS
            ;3115  ALU_D[AND]R[0A],   ; See if BIT 0 of INFO_SYMBOLS is 0
U 12F7. 000D,0034,0180,0A50,0010,12F8  ;3116    CLK.UBCC  ; ...
U 12F8. 0800,013C,0180,0A50,0000,1154  ;3117  Z?,D_R[0A]   ; Get SYMBOL_POINTER
U 1154. 001B,001C,1980,0988,0000,1155  ;3118 =0 RC[1]_NOT.K[ZERO]  ; BIT 0 was NOT 0, set CODE_VECTOR to
            ;3119       ; ...all ones
U 1155. 0021,003C,0180,0AD0,0000,1156  ;3120  R[0A]_D.LEFT,SI/ZERO  ; Shift SYMBOL_POINTER left one bit
            ;3121       ; ...pointing to the next bit to
            ;3122       ; ...be encoded
U 1156. 0000,003D,0180,0800,0000,12FB  ;3123 =0 CALL[FIRST.ORDER]  ; Start encoding the first order
            ;3124       ; ...symbols into the final vector
U 1157. 0000,003C,0180,0800,0000,1158  ;3125  NOP
U 1158. 0000,003D,0180,0800,0000,1307  ;3126 =0 CALL[SECOND.ORDER]  ; Start encoding the second order
            ;3127       ; ...symbols into the final vector
U 1159. 0000,003E,0180,0800,0000,0001  ;3128  RETURN1    ; Return to caller with the encoded
            ;3129       ; ...vector in RC[1]
            ;3130
            ;3131
            ;3132
```

```
            ;3133  .PAGE
            ;3134  .TOC "    REED-MULLER VECTOR GENERATION ROUTINE"
            ;3135 ;*****************************************************************
            ;3136 ;*
            ;3137 ; THIS SUBROUTINE GENERATES THE VECTORS USED BY THE
            ;3138 ; REED-MULLER ALGORITHM TO GENERATE CODEWORDS.  THESE
            ;3139 ; VECTORS ARE:
            ;3140 ;
            ;3141 ;  V1: 10101010101010101010101010101010
            ;3142 ;  V2: 11001100110011001100110011001100
            ;3143 ;  V3: 11110000111100001111000011110000
            ;3144 ;  V4: 11111111000000001111111100000000
            ;3145 ;  V5: 11111111111111110000000000000000
            ;3146 ;
            ;3147 ; THE ROUTINE IS ARRANGED IN A TABLE, IN ORDER TO
            ;3148 ; BRANCH INTO IT WITH D3-0? BRANCH.
            ;3149 ;
            ;3150 ; **NOTE**: D<3:0> MUST HAVE THE VECTOR NUMBER WHEN
            ;3151 ;    CALLING THE ROUTINE.
            ;3152 ;
            ;3153 ; THE VECTOR IS RETURNED IN THE D-REGISTER.
            ;3154 ;
            ;3155 ; SIDE EFFECTS:
            ;3156 ;
            ;3157 ; The contents of the following registers will be lost:
            ;3158 ;
            ;3159 ; Q, SC
            ;3160 ;-
            ;3161 GENERATE.VECTOR:
U 12F9, 0500,003C,0180,0800,0000,12FA  ;3162  D_D.LEFT,SI/ZERO  ; Multiply VECTOR Pointer by 2
            ;3163       ; ... so as to index into the table
            ;3164       ; ...below
U 12FA, 0000,193C,0180,0800,0000,1060  ;3165  D3-0?    ; Index into the Vector table
            ;3166 =0000 D_NOT.K[ZERO],   ; Generate V(0)
U 1060, 081B,001E,1980,0800,0000,0001  ;3167   RETURN1   ; ...
U 1061, 0000,003C,0180,0800,0000,1062  ;3168  NOP
U 1062, 001B,0000,1180,0800,0082,1063  ;3169 V1: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1063, 0F1B,0010,11C0,0800,0000,106C  ;3170  ALU_0+K[.4]+1,Q_ALU,D_0,J/VSHF  ; Q<3:0>=0101
U 1064, 001B,0000,1180,0800,0082,1065  ;3171 V2: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1065, 0F18,0038,0DC0,0800,0000,106C  ;3172  Q_K[.33],D_0,J/VSHF  ; Q<3:0>=0011
U 1066, 001B,0000,0180,0800,0082,1067  ;3173 V3: ALU_0-K[.8],SC_ALU  ; SET SHIFT VALUE = -8
U 1067, 0F18,0038,61C0,0800,0000,106C  ;3174  Q_K[.F],D_0,J/VSHF  ; Q<7:0>=00001111
U 1068, 001B,001C,6580,0800,0082,1069  ;3175 V4: ALU_NOT.K[.10],SC_ALU  ; K[SLOW],SHIFT VALUE = -16.
U 1069, 0F18,0038,49C0,0800,0080,D06C  ;3176  Q_K[.FF],D_0,SC_SC+1,J/VSHF ; Q<7:0>=11111111
U 106A, 0818,0038,C180,0800,0000,106F  ;3177 V5: D_K[.FFFF],J/V.EXIT
U 106B, 0000,003C,0180,0800,0000,106C  ;3178  NOP
U 106C, 0D00,003C,0180,0800,0000,106D  ;3179 VSHF: D_DAL.SC
U 106D, 0000,193C,0180,0800,0000,106E  ;3180  D0?    ; ANY V(I) IS DONE WHEN D<0>=1
U 106E, 0000,003C,0180,0800,0000,106C  ;3181 =1110 J/VSHF    ; LUCKY BR CONSTRAINT
            ;3182 V.EXIT:
            ;3183  D_NOT.D,   ; Adjust D as necessary
U 106F, 0801,002A,0180,0800,0000,0001  ;3184   RETURN1   ; Return to caller
            ;3185
```

```
;3186  .PAGE
;3187  .TOC "   FIRST ORDER REED-MULLER ENCODE"
;3188 ;*************************************************************
;3189 ;**
;3190 ;
;3191 ; Functional Description:
;3192 ; ----------------------
;3193 ;
;3194 ;      This subroutine is used by the ENCODE and DECODE
;3195 ; routines to encode the first order Reed-Muller vector
;3196 ; bits <05:01> of the received vector.
;3197 ;
;3198 ;
;3199 ; Calling Constraint: =0
;3200 ; ------------------
;3201 ;
;3202 ;
;3203 ; Inputs:
;3204 ; ------
;3205 ;
;3206 ; RC[2] must hold the vector to be encoded (INFO_SYMBOLS)
;3207 ;
;3208 ; R[0A] must hold the SYMBOL_POINTER as passed by
;3209 ; subroutines ENCODE or DECODE
;3210 ;
;3211 ; RC[1] must hold the CODE_VECTOR as passed by
;3212 ; subroutines ENCODE or DECODE
;3213 ;
;3214 ; Outputs:
;3215 ; -------
;3216 ;
;3217 ; RC[1] will contain upon exit the CODE_VECTOR for
;3218 ; bits <05:01> of INFO_SYMBOLS
;3219 ;
;3220 ; R[0A] will have an UPDATED SYMBOL_POINTER
;3221 ;
;3222 ;
;3223 ; Side Effects:
;3224 ; ------------
;3225 ;
;3226 ; The contents of the following registers will be lost
;3227 ;
;3228 ; D, Q, R[0], R[2]
;3229 ;
;3230 ;--
;3231 ;*************************************************************
;3232 FIRST.ORDER:
U 12FB, 0018,0038,0980,0A80,0000,12FC  ;3233  R[0]_K[.2]    ; Initialize the COUNTER
U 12FC, 0018,0038,0580,0A90,0000,12FD  ;3234  R[2]_K[.1]    ; Initialize variable J
;3235 FRST.ORDR.LOOP:
U 12FD, 0810,0038,0180,0910,0000,12FE  ;3236  D_RC[2]      ; Get INFO_SYMBOLS
;3237  ALU_D[AND]R[0A],   ; See if bit pointed by SYMBOL_POINTER
U 12FE, 000D,0034,0180,0A50,0010,12FF  ;3238   CLK.UBCC    ; ...in INFO_SYMBOLS is set
U 12FF, 0000,013C,0180,0800,0000,115A  ;3239  Z?
U 115A, 0000,003C,0180,0800,0000,115C  ;3240 =0 J/FRST.ORDR.SYM.NOT0  ; Bit was not 0
```

```
U 115B, 0000,003C,0180,0800,0000,1301  ;3241  J/FRST.ORDR.SYM.EQ0  ; Bit was 0
          ;3242 =0
          ;3243 FRST.ORDR.SYM.NOT0:
          ;3244  D_R[2],     ; Point to VECTOR(J)
U 115C, 0800,003D,0180,0A10,0000,12F9  ;3245  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 115D, 0010,0038,01C0,0908,0000,1300  ;3246  Q_RC[1]     ; Get CODE_VECTOR
          ;3247  ALU_D.XOR.Q,    ; Encode current bit pointed by
U 1300, 001D,0020,0180,0988,0000,1301  ;3248   RC[1]_ALU  ; ...SYMBOL_POINTER and save in RC[1]
          ;3249 FRST.ORDR.SYM.EQ0:
U 1301, 0800,003C,0180,0A10,0000,1302  ;3250  D_R[2]     ; Get J
U 1302, 0019,0014,0580,0A90,0000,1303  ;3251  R[2]_D+K[.1]    ; Increment J
U 1303, 0800,003C,0180,0A50,0000,1304  ;3252  D_R[0A]     ; Get SYMBOL_POINTER
U 1304, 0021,003C,D580,0AD0,0000,1305  ;3253  R[0A]_D.LEFT,SI/ZERO,K[.6] ; Point to the next bit to be encoded
          ;3254  ALU_R[0][XOR]K[.6],   ; Is the COUNTER equal to 6?
U 1305, 0018,0020,D580,0A00,0010,1306  ;3255   CLK.UBCC  ; ...(i.e., COUNTER .LE. 5)
          ;3256  Z?,     ;
U 1306, 0800,013C,0180,0A00,0000,115E  ;3257   D_R[0]   ; Get the COUNTER just in case it has
          ;3258        ; ...to be incremented
          ;3259 =0 J/FRST.ORDR.LOOP,  ; Counter is NOT equal to 6 yet, thus
U 115E, 0019,0014,0580,0A80,0000,12FD  ;3260   R[0]_D+K[.1]  ; ...we did not encode all First
          ;3261        ; ...Order Symbols.  Repeat the loop
U 115F, 0000,003E,0180,0800,0000,0001  ;3262  RETURN1    ; OK. COUNTER is equal to 6 so
          ;3263        ; ...we finished encoding ALL First
          ;3264        ; ...Order Symbols.  Return to caller.
          ;3265
          ;3266
```

```
                ;3267  .PAGE
                ;3268  .TOC "    SECOND ORDER REED-MULLER ENCODING ROUTINE"
                ;3269 ;*************************************************************
                ;3270 ;**
                ;3271 ;
                ;3272 ; Functional Description:
                ;3273 ; ----------------------
                ;3274 ;
                ;3275 ;      This subroutine is used by the ENCODE and DECODE
                ;3276 ; routines to encode the Second order Reed-Muller vector
                ;3277 ; of INFO_SYMBOLS vector.
                ;3278 ;
                ;3279 ;
                ;3280 ; Calling Constraint: =0
                ;3281 ; ------------------
                ;3282 ;
                ;3283 ;
                ;3284 ; Inputs:
                ;3285 ; ------
                ;3286 ;
                ;3287 ; RC[1] must hold the CODE_VECTOR as set up by FIRST.ORDER
                ;3288 ; routine or by the DECODE routine.
                ;3289 ;
                ;3290 ; RC[2] must hold the INFO_SYMBOLS vector (to be encoded)
                ;3291 ;
                ;3292 ; R[0A] must hold the SYMBOL_POINTER as set up by FIRST.ORDER
                ;3293 ; routine or by the DECODE routine.
                ;3294 ;
                ;3295 ;
                ;3296 ; Outputs:
                ;3297 ; -------
                ;3298 ;
                ;3299 ; RC[1] will hold upon exit the encoded vector
                ;3300 ;
                ;3301 ;
                ;3302 ; Side Effects:
                ;3303 ; ------------
                ;3304 ;
                ;3305 ; The contents of the following registers will be lost:
                ;3306 ;
                ;3307 ; D, Q, RC[1], R[0] through R[2], R[0A], SC
                ;3308 ;
                ;3309 ;--
                ;3310 ;*************************************************************
                ;3311 SECOND.ORDER:
U 1307, 0018,0038,0580,0A80,0000,1308  ;3312  R[0]_K[.1]    ; Initialize COUNTER to 1
U 1308, 0F18,0038,1180,0A90,0000,1309  ;3313  R[2]_K[.4],D_0    ; Initialize variable J to 4
U 1309, 0019,0010,1180,0A88,0000,130A  ;3314  ALU_D+K[.4]+1,R[1]_ALU  ; Initialize variable I to 5
                ;3315 SCND.ORDR.LOOP:
U 130A, 0810,0038,0180,0910,0000,130B  ;3316  D_RC[2]     ; Get INFO_SYMBOLS
                ;3317  ALU_D[AND]R[0A],  ; See if bit pointed by SYMBOL_POINTER
U 130B, 000D,0034,0180,0A50,0010,130C  ;3318   CLK.UBCC  ; ...in INFO_SYMBOLS is set
U 130C, 0000,013C,0180,0800,0000,1160  ;3319  Z?
U 1160, 0000,003C,0180,0800,0000,1162  ;3320 =0 J/SCND.ORDR.SYM.NOT0  ; Bit was not 0
U 1161, 0000,003C,0180,0800,0000,130F  ;3321  J/SCND.ORDR.SYM.EQ0  ; Bit was 0
```

```
            ;3322 =0
            ;3323 SCND.ORDR.SYM.NOT0:
            ;3324  D_R[1],     ; Point to VECTOR(I)
U 1162. 0800.003D.0180.0A08.0000.12F9  ;3325  CALL[GENERATE.VECTOR]  ; ...and call routine to
U 1163. 0001.003C.0180.0AF0.0000.1164  ;3326  R[0E]_D     ; Save VECTOR(I)
            ;3327 =0 D_R[2],     ; Point to VECTOR(J)
U 1164. 0800.003D.0180.0A10.0000.12F9  ;3328  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 1165. 0000.003C.01C0.0A70.0000.130D  ;3329  Q_R[0E]     ; Get saved VECTOR(I)
            ;3330  D_D.AND.Q,    ; "Combine" VECTOR(I) with VECTOR(J)
U 130D. 081D.0034.0180.0908.0000.130E  ;3331    LC_RC[1]  ; Get CODE_VECTOR
U 130E. 0011.0020.0180.0988.0000.130F  ;3332  RC[1]_D.XOR.LC    ; Encode bit pointed by SYMBOL_POINTER
            ;3333       ; ...in CODE_VECTOR
            ;3334 SCND.ORDR.SYM.EQ0:
            ;3335  ALU_R[2][A-B]K[.1],D_ALU, ; Decrement J
U 130F. 0818.0000.0580.0A10.0010.1310  ;3336   CLK.UBCC  ; Is it 0 yet?
            ;3337  Z?,
U 1310. 0000.013C.01C0.0A08.0000.1166  ;3338   Q_R[1]    ; Get I
            ;3339 =0 J/SCND.ORDR.JNT0,   ; J is NOT 0.
U 1166. 0001.003C.0180.0A90.0000.1312  ;3340   R[2]_D    ; Save J
U 1167. 0019.2000.0580.0A90.0000.1311  ;3341  R[2]_Q-K[.1]    ; Decrement I and save
U 1311. 0019.2000.0580.0A88.0000.1312  ;3342  R[1]_Q-K[.1]    ; Reinitialize J to I - 1
            ;3343 SCND.ORDR.JNT0:
U 1312. 0800.003C.0180.0A50.0000.1313  ;3344  D_R[0A]     ; Get SYMBOL_POINTER
U 1313. 0021.003C.F5F8.0AD0.0000.1314  ;3345  R[0A]_D.LEFT,SI/ZERO,Q_0,K[.A] ; Point to the next bit to be encoded
U 1314. 0819.2010.F580.0800.0000.1315  ;3346  ALU_Q+K[.A]+1,D_ALU  ; D = d11
            ;3347  ALU_D.XOR.R[0],    ; Is the COUNTER equal to d11?
U 1315. 000D.0020.0180.0A00.0010.1316  ;3348   CLK.UBCC  ; ...
            ;3349  Z?,      ; ...
U 1316. 0800.013C.0180.0A00.0000.1168  ;3350   D_R[0]    ; Get the COUNTER in case it is
            ;3351       ; ...not equal to d11
            ;3352 =0 J/SCND.ORDR.LOOP,   ; COUNTER is NOT equal to d11 yet
U 1168. 0019.0014.0580.0A80.0000.130A  ;3353   R[0]_D+K[.1]  ; Increment and save the COUNTER
U 1169. 0000.003E.0180.0800.0000.0001  ;3354  RETURN1     ; Return to caller.
            ;3355       ; ...Finished encoding all second
            ;3356       ; ...order terms
            ;3357
```

```
                ;3358  .PAGE
                ;3359  .TOC "   REED-MULLER DECODE ROUTINE"
                ;3360 ;*************************************************************
                ;3361 ;**
                ;3362 ;
                ;3363 ; Functional Description:
                ;3364 ; ----------------------
                ;3365 ;
                ;3366 ;      This subroutine is used to decode the Reed-Muller
                ;3367 ; codes encoded by the ENCODE routine.
                ;3368 ;
                ;3369 ; Calling Constraint: =00
                ;3370 ; ------------------
                ;3371 ;
                ;3372 ;
                ;3373 ; Inputs:
                ;3374 ; ------
                ;3375 ;
                ;3376 ; RC[0] must hold the 32 bit vector to be decoded
                ;3377 ;
                ;3378 ;
                ;3379 ; Outputs:
                ;3380 ; -------
                ;3381 ;
                ;3382 ; RC[2] will hold the decoded vector
                ;3383 ;
                ;3384 ;
                ;3385 ; Side Effects:
                ;3386 ; ------------
                ;3387 ;
                ;3388 ;
                ;3389 ; See header of subroutine ENCODE
                ;3390 ;
                ;3391 ;
                ;3392 ;--
                ;3393 ;*************************************************************
                ;3394 DECODE:
U 1317. 0018.0038.6580.0A98.0000.1318  ;3395  R[3]_K[.10]    ; Initialize variable K to d16
U 1318. 0018.0038.6580.0AA0.0000.1319  ;3396  R[4]_K[.10]    ; Initialize variable L to d16
U 1319. 0003.003C.0180.0990.0000.131A  ;3397  RC[2]_0      ; Initialize variable INFO_SYMBOLS to 0
U 131A. 0810.0038.0180.0900.0000.131B  ;3398  D_RC[0]      ; Get the RCVD_VECTOR
U 131B. 0001.003C.0180.0AB0.0000.131C  ;3399  R[6]_D     ; Save RCVD_VECTOR
                ;3400 DECODE.SCND.ORDR.LOOP:
U 131C. 0840.003C.0180.0A20.0000.131D  ;3401  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; Divide L by 2
U 131D. 0001.003C.0180.0AA0.0000.131E  ;3402  R[4]_D     ; L = L/2
U 131E. 0003.003C.0180.0AC0.0000.131F  ;3403  R[8]_0     ; Initialize variable 1ST to 0
U 131F. 0003.003C.0180.0AC8.0000.1320  ;3404  R[9]_0     ; Initialize variable SAVE_1ST to 0
U 1320. 0003.003C.0180.0998.0000.1321  ;3405  RC[3]_0      ; Initialize varaible COVERED_BITS to 0
U 1321. 0018.0038.0580.0A80.0000.1322  ;3406  R[0]_K[.1]    ; Initialize COUNTER to 1
U 1322. 0003.003C.0180.0AB8.0000.1323  ;3407  R[7]_0     ; Initialize variable ONE_COUNTER to 0
                ;3408 DECODE.SEARCH.LOOP:
U 1323. 0003.003C.0180.0AA8.0000.1324  ;3409  R[5]_0     ; Initialize variable SUM to 0
U 1324. 0003.003C.0180.0AD8.0000.1325  ;3410  R[0B]_0     ; Initialize variable CURRENT_BIT_MASK
                ;3411       ; ...to 0
U 1325. 0000.003C.0180.0A40.0082.1326  ;3412  SC_R[8]     ; Get 1ST
```

```
U 1326. 0003.001C.0180.0AD8.0000.1327  ;3413  R[0B]_NOT.MASK   ; Set bit 1ST in CURRENT_BIT_MASK
        ;3414  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST set in RCVD_VECTOR?
U 1327. 0000.0024.0180.0A30.0010.1328  ;3415   CLK.UBCC  ; ...
U 1328. 0800.013C.0180.0A28.0000.116A  ;3416  Z?,D_R[5]   ; Get SUM
U 116A. 0019.0014.0580.0AA8.0000.116B  ;3417 =0 R[5]_D+K[.1]   ; Bit 1ST in RCVD_VECTOR was NOT 0,
        ;3418       ; ...increment the SUM
U 116B. 0800.003C.0180.0A40.0000.1329  ;3419  D_R[8]    ; Get 1ST
U 1329. 000D.0014.01C0.0A20.0082.132A  ;3420  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 132A. 0800.001C.0180.0A58.0000.132B  ;3421  D_R[0B].ORNOT.MASK  ; Set bit 1ST + L in CURRENT_BIT_MASK
U 132B. 0001.003C.0180.0AD8.0000.132C  ;3422  R[0B]_D    ; Save CURRENT_BIT_MASK
        ;3423  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + L set in RCVD_VECTOR?
U 132C. 0000.0024.0180.0A30.0010.132D  ;3424   CLK.UBCC  ; ...
U 132D. 0800.013C.0180.0A28.0000.116C  ;3425  Z?,D_R[5]   ; Get SUM
U 116C. 0019.0014.0580.0AA8.0000.116D  ;3426 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 116D. 0800.003C.0180.0A58.0000.132E  ;3427  D_R[0B]    ; Get CURRENT_BIT_MASK
U 132E. 000D.2014.0180.0A18.0082.132F  ;3428  ALU_Q[A+B]R[3],SC_ALU  ; SC = 1ST + K + L
U 132F. 0001.001C.0180.0AD8.0000.1330  ;3429  R[0B]_D.ORNOT.MASK  ; Set bit 1ST + K + L in
        ;3430       ; ...CURRENT_BIT_MASK
        ;3431  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + K + L set
U 1330. 0000.0024.0180.0A30.0010.1331  ;3432   CLK.UBCC  ; ...in RCVD_VECTOR?
U 1331. 0800.013C.0180.0A28.0000.116E  ;3433  Z?,D_R[5]   ; Get SUM
U 116E. 0019.0014.0580.0AA8.0000.116F  ;3434 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 116F. 0800.003C.0180.0A40.0000.1332  ;3435  D_R[8]    ; Get 1ST
U 1332. 000D.0014.0180.0A18.0082.1333  ;3436  ALU_D[A+B]R[3],SC_ALU  ; SC = 1ST + K
U 1333. 0800.001C.0180.0A58.0000.1334  ;3437  D_R[0B].ORNOT.MASK  ; Set bit 1ST + K in CURRENT_BIT_MASK
U 1334. 0001.003C.0180.0AD8.0000.1335  ;3438  R[0B]_D    ; Save CURRENT_BIT_MASK
        ;3439  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + K set in RCVD_VECTOR
U 1335. 0000.0024.0180.0A30.0010.1336  ;3440   CLK.UBCC  ; ...
U 1336. 0800.013C.0180.0A28.0000.1170  ;3441  Z?,D_R[5]   ; Get SUM
U 1170. 0019.0014.0580.0AA8.0000.1171  ;3442 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 1171. 0800.003C.0180.0A58.0000.1337  ;3443  D_R[0B]    ; Get CURRENT_BIT_MASK
        ;3444  ALU_D[AND]RC[3],  ; Are bits set in CURRENT_BIT_MASK
U 1337. 0011.0034.0180.0918.0010.1338  ;3445   CLK.UBCC  ; ...also set in COVERED_BITS
U 1338. 0000.013C.0180.0800.0000.1172  ;3446  Z?
        ;3447 =0 J/DECODE.BITS.COVERED,  ; Some of the bits in CURRENT_BIT_MASK
U 1172. 0820.003C.0180.0A20.0000.1339  ;3448  D_R[4].LEFT.SI/ZERO  ; ...have been previously covered.
        ;3449       ; ...D = 2 * L
        ;3450  J/DECODE.BITS.NOT.COVERED, ; Bits in CURRENT_BIT_MASK were not
U 1173. 0818.0014.0580.0A40.0000.133C  ;3451  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
        ;3452 DECODE.BITS.COVERED:
U 1339. 080D.0014.0180.0A48.0000.133A  ;3453  ALU_D[A+B]R[9],D_ALU  ; D = SAVE_1ST + (2 * L)
U 133A. 0001.003C.0180.0AC0.0000.133B  ;3454  R[8]_D    ; 1ST = SAVE_1ST + (2 * L)
        ;3455  R[9]_D,    ; SAVE_1ST = SAVE_1ST + (2 * L)
U 133B. 0001.003C.0180.0AC8.0000.1323  ;3456   J/DECODE.SEARCH.LOOP ; Repeat the search loop for
        ;3457       ; ...unCOVERED_BITS
        ;3458 DECODE.BITS.NOT.COVERED:
U 133C. 0001.003C.0180.0AC0.0000.133D  ;3459  R[8]_D    ; 1ST = 1ST + 1
U 133D. 0800.003C.0180.0A58.0000.133E  ;3460  D_R[0B]    ; Get CURRENT_BIT_MASK
U 133E. 0811.0030.0180.0918.0000.133F  ;3461  D_D.OR.RC[3]   ; COVERED_BITS = COVERED_BITS .OR.
        ;3462       ; ...CURRENT_BIT_MASK
U 133F. 0001.003C.0180.0998.0000.1340  ;3463  RC[3]_D    ; Save COVERED_BITS mask
U 1340. 0000.003C.01C0.0A38.0000.1341  ;3464  Q_R[7]    ; Get ONE_COUNTER
U 1341. 0800.003C.0180.0A28.0000.1342  ;3465  D_R[5]    ; Get SUM and see if it is EVEN or ODD
U 1342. 0000.193C.0180.0800.0000.107E  ;3466  D1-0?    ; ...
U 107E. 0000.003C.0180.0800.0000.1343  ;3467 =1110 J/DECODE.SUM.EVEN  ; SUM was even
```

```
U 107F, 0019,2014,0580,0AB8,0000,1343  ;3468  R[7]_Q+K[.1]   ; SUM is ODD.  Increment ONE_COUNTER
        ;3469 DECODE.SUM.EVEN:
        ;3470  ALU_R[0].XOR.K[.8],  ; Is COUNTER equal to 8?
U 1343, 0018,0020,0180,0A00,0010,1344  ;3471   CLK.UBCC  ; ...
U 1344, 0800,013C,0180,0A00,0000,1174  ;3472  Z?,D_R[0]   ; Get COUNTER in case it has to be
        ;3473       ; ...incremented
        ;3474 =0 J/DECODE.SEARCH.LOOP,  ; COUNTER is NOT equal to 8 yet
U 1174, 0019,0014,0580,0A80,0000,1323  ;3475   R[0]_D+K[.1]  ; Increment COUNTER
        ;3476  ALU_R[7][A-B]K[.4],  ; Is ONE_COUNTER GREATER than or
U 1175, 0018,0000,1180,0A38,0010,1345  ;3477   CLK.UBCC  ; ...EQUAL to  4?
        ;3478  ALU.N?,    ; Check Condition Codes.
        ;3479   D_RC[2],  ; Get INFO_SYMBOLS
U 1345, 0810,1B38,4580,0910,0000,1047  ;3480   K[.8000]  ; Bring up slow constant
U 1047, 0819,0030,4580,0800,0000,104F  ;3481 =0111 D_D.OR.K[.8000]   ; ONE_COUNTER GREATER than or
        ;3482       ; ...EQUAL to 4.  Set bit 0 of
        ;3483       ; ...INFO_SYMBOLS
        ;3484  RC[2]_ALU.RIGHT,SI/ZERO, ; Shift INFO_SYMBOLS right and Save
U 104F, 0041,003C,0180,0990,0000,1346  ;3485   ALU_D   ; ...
        ;3486  ALU_R[4][XOR]K[.1],  ; Is L EQUAL to 1?
U 1346, 0018,0020,0580,0A20,0010,1347  ;3487   CLK.UBCC  ; ...
        ;3488  Z?,
U 1347, 0840,013C,0180,0A18,0000,1176  ;3489  D_ALU.RIGHT,ALU_R[3],SI/ZERO ; D = K/2. In case L = 1
U 1176, 0000,003C,0180,0800,0000,131C  ;3490 =0 J/DECODE.SCND.ORDR.LOOP  ; L was not EQUAL to 1. Not all SECOND
        ;3491       ; ...ORDER symbols have been decoded.
        ;3492       ; ...Repeat loop.
        ;3493  ALU_D.XOR.K[.1],  ; Is K = 1?
U 1177, 0019,0020,0580,0800,0010,1348  ;3494   CLK.UBCC  ; ...
U 1348, 0001,013C,0180,0A98,0000,1178  ;3495  Z?,R[3]_D   ; Save K just in case.
        ;3496 =0 J/DECODE.SCND.ORDR.LOOP, ; OK. K was updated and it is NOT equal
U 1178, 0001,003C,0180,0AA0,0000,131C  ;3497   R[4]_D   ; ...to 1.  Update L.  L = K/2
U 1179, 0810,0038,0180,0910,0000,1349  ;3498  D_RC[2]    ; Get INFO_SYMBOLS
U 1349, 0200,003C,0180,0800,0000,134A  ;3499  D_D.RIGHT2,SI/ZERO  ; Shift INFO_SYMBOLS right 4 bits
U 134A, 0200,003C,0180,0800,0000,134B  ;3500  D_D.RIGHT2,SI/ZERO  ; ...
U 134B, 0001,003C,0180,0990,0000,134C  ;3501  RC[2]_D    ; Save INFO_SYMBOLS
U 134C, 0018,0038,0980,0AD0,0000,134D  ;3502  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 134D, 0003,003C,0180,0988,0000,117A  ;3503  RC[1]_0    ; Clear CODE_VECTOR
U 117A, 0000,003D,0180,0800,0000,1307  ;3504 =0 CALL[SECOND.ORDER]  ; Re-encode the second order symbols
        ;3505       ; ...so as to extract them from
        ;3506       ; ...RCVD_VECTOR
U 117B, 0800,003C,0180,0A30,0000,134E  ;3507  D_R[6]    ; Get RCVD_VECTOR
U 134E, 0811,0020,0180,0908,0000,134F  ;3508  D_D.XOR.RC[1]   ; "Subtract" the encoded second order
        ;3509       ; ...symbols from RCVD_VECTOR
U 134F, 0001,003C,0180,0AB0,0000,1350  ;3510  R[6]_D    ; Save the result
U 1350, 0018,0038,7580,0AA0,0000,1351  ;3511  R[4]_K[.20]   ; Initialize variable L to d32
        ;3512 DECODE.FRST.ORDER.LOOP:
U 1351, 0840,003C,0180,0A20,0000,1352  ;3513  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; D = L/2
U 1352, 0001,003C,0180,0AA0,0000,1353  ;3514  R[4]_D    ; Save L
U 1353, 0003,003C,0180,0AC0,0000,1354  ;3515  R[8]_0    ; Initialize variable 1ST
U 1354, 0003,003C,0180,0AC8,0000,1355  ;3516  R[9]_0    ; Initialize variable SAVE_1ST
U 1355, 0003,003C,0180,0998,0000,1356  ;3517  RC[3]_0    ; Initialize variable COVERED_BITS to 0
U 1356, 0018,0038,0580,0A80,0000,1357  ;3518  R[0]_K[.1]   ; Initialize COUNTER to 1
U 1357, 0003,003C,0180,0AB8,0000,1358  ;3519  R[7]_0    ; Clear ONE_COUNTER
        ;3520 DECODE.FRST.ORDR.SEARCH:
U 1358, 0003,003C,0180,0AA8,0000,1359  ;3521  R[5]_0    ; Initialize variable SUM to 0
U 1359, 0003,003C,0180,0AD8,0000,135A  ;3522  R[0B]_0    ; Initialize variable CURRENT_BIT_MASK
```

```
            ;3523       ; ...to 0
U 135A, 0000.003C.0180.0A40.0082.135B  ;3524  SC_R[8]     ; Get 1ST
U 135B, 0003.001C.0180.0AD8.0000.135C  ;3525  R[0B]_NOT.MASK    ; Set bit 1ST in CURRENT_BIT_MASK
            ;3526  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST set in RCVD_VECTOR?
U 135C, 0000.0024.0180.0A30.0010.135D  ;3527   CLK.UBCC   ; ...
U 135D, 0800.013C.0180.0A28.0000.117C  ;3528  Z?,D_R[5]    ; Get SUM
U 117C, 0019.0014.0580.0AA8.0000.117D  ;3529 =0 R[5]_D+K[.1]    ; Bit 1ST in RCVD_VECTOR was NOT 0,
            ;3530       ; ...increment the SUM
U 117D, 0800.003C.0180.0A40.0000.135E  ;3531  D_R[8]     ; Get 1ST
U 135E, 000D.0014.01C0.0A20.0082.135F  ;3532  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 135F, 0800.001C.0180.0A58.0000.1360  ;3533  D_R[0B].ORNOT.MASK   ; Set bit 1ST + L in CURRENT_BIT_MASK
U 1360, 0001.003C.0180.0AD8.0000.1361  ;3534  R[0B]_D     ; Save CURRENT_BIT_MASK
            ;3535  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + L set in RCVD_VECTOR?
U 1361, 0000.0024.0180.0A30.0010.1362  ;3536   CLK.UBCC   ; ...
U 1362, 0800.013C.0180.0A28.0000.117E  ;3537  Z?,D_R[5]    ; Get SUM
U 117E, 0019.0014.0580.0AA8.0000.117F  ;3538 =0 R[5]_D+K[.1]    ; Bit was NOT 0. Increment SUM
U 117F, 0800.003C.0180.0A58.0000.1363  ;3539  D_R[0B]     ; Get CURRENT_BIT_MASK
            ;3540  ALU_D[AND]RC[3],    ; Are bits set in CURRENT_BIT_MASK
U 1363, 0011.0034.0180.0918.0010.1364  ;3541   CLK.UBCC   ; ...also set in COVERED_BITS
U 1364, 0000.013C.0180.0800.0000.1180  ;3542  Z?
            ;3543 =0 J/DECODE.BITS.COVRD,   ; Some of the bits in CURRENT_BIT_MASK
U 1180, 0820.003C.0180.0A20.0000.1365  ;3544  D_R[4].LEFT,SI/ZERO   ; ...have been previously covered.
            ;3545       ; ...D = 2 * L
            ;3546  J/DECODE.BITS.NOT.COVRD, ; Bits in CURRENT_BIT_MASK were not
U 1181, 0818.0014.0580.0A40.0000.1368  ;3547  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
            ;3548 DECODE.BITS.COVRD:
U 1365, 080D.0014.0180.0A48.0000.1366  ;3549  ALU_D[A+B]R[9],D_ALU   ; D = SAVE_1ST + (2 * L)
U 1366, 0001.003C.0180.0AC0.0000.1367  ;3550  R[8]_D     ; 1ST = SAVE_1ST + (2 * L)
            ;3551  R[9]_D,     ; SAVE_1ST = SAVE_1ST + (2 * L)
U 1367, 0001.003C.0180.0AC8.0000.1358  ;3552     J/DECODE.FRST.ORDR.SEARCH ; Repeat the search loop for
            ;3553       ; ...unCOVERED_BITS
            ;3554 DECODE.BITS.NOT.COVRD:
U 1368, 0001.003C.0180.0AC0.0000.1369  ;3555  R[8]_D     ; 1ST = 1ST + 1
U 1369, 0800.003C.0180.0A58.0000.136A  ;3556  D_R[0B]     ; Get CURRENT_BIT_MASK
U 136A, 0811.0030.0180.0918.0000.136B  ;3557  D_D.OR.RC[3]    ; COVERED_BITS = COVERED_BITS .OR.
            ;3558       ; ...CURRENT_BIT_MASK
U 136B, 0001.003C.0180.0998.0000.136C  ;3559  RC[3]_D     ; Save COVERED_BITS mask
U 136C, 0000.003C.01C0.0A38.0000.136D  ;3560  Q_R[7]     ; Get ONE_COUNTER
U 136D, 0800.003C.0180.0A28.0000.136E  ;3561  D_R[5]     ; Get SUM and see if it is EVEN or ODD
U 136E, 0000.193C.0180.0800.0000.108E  ;3562  D1-0?      ; ...
U 108E, 0000.003C.0180.0800.0000.136F  ;3563 =1110 J/DECODE.SUM.EVEN2   ; SUM was even
U 108F, 0019.2014.0580.0AB8.0000.136F  ;3564  R[7]_Q+K[.1]    ; SUM is ODD.  Increment ONE_COUNTER
            ;3565 DECODE.SUM.EVEN2:
            ;3566  ALU_R[0].XOR.K[.10],   ; Is COUNTER equal to d16?
U 136F, 0018.0020.6580.0A00.0010.1370  ;3567   CLK.UBCC   ; ...
U 1370, 0800.013C.0180.0A00.0000.1182  ;3568  Z?,D_R[0]    ; Get COUNTER in case it has to be
            ;3569       ; ...incremented
            ;3570 =0 J/DECODE.FRST.ORDR.SEARCH, ; COUNTER is NOT equal to 16 yet
U 1182, 0019.0014.0580.0A80.0000.1358  ;3571   R[0]_D+K[.1]   ; Increment COUNTER
            ;3572  ALU_R[7][A-B]K[.8],   ; Is ONE_COUNTER GREATER than or
U 1183, 0018.0000.0180.0A38.0010.1371  ;3573   CLK.UBCC   ; ...EQUAL to  8?
            ;3574  ALU.N?,     ; Check Condition Codes.
U 1371, 0810.1B38.0180.0910.0000.1057  ;3575   D_RC[2]    ; Get INFO_SYMBOLS
U 1057, 0819.0030.0580.0800.0000.105F  ;3576 =0111 D_D.OR.K[.1]    ; ONE_COUNTER GREATER than or
            ;3577       ; ...EQUAL to 8.  Set bit 0 of
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1005
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58                   Page   9

          ;3578       ; ...INFO_SYMBOLS
U 105F, 0021,003C,0180,0990,0000,1372  ;3579  RC[2]_D.LEFT,SI/ZERO  ; Shift INFO_SYMBOLS left and Save
          ;3580  ALU_R[4].XOR.K[.1],  ; Is L = 1?
U 1372, 0018,0020,0580,0A20,0010,1373  ;3581   CLK.UBCC
U 1373, 0000,013C,0180,0800,0000,1184  ;3582  Z?
U 1184, 0000,003C,0180,0800,0000,1351  ;3583 =0 J/DECODE.FRST.ORDER.LOOP ; L is NOT equal to 1 yet. Repeat loop.
U 1185, 0003,003C,0180,0988,0000,1374  ;3584  RC[1]_0     ; Initialize CODE_VECTOR to 0
U 1374, 0018,0038,0980,0AD0,0000,1186  ;3585  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 1186, 0000,003D,0180,0800,0000,12FB  ;3586 =0 CALL[FIRST.ORDER]  ; Generate the first order Reed-Muller
          ;3587       ; ...code for the symbols just
          ;3588       ; ...decoded
U 1187, 0800,003C,0180,0A30,0000,1375  ;3589  D_R[6]     ; Get RCVD_VECTOR
U 1375, 0010,0038,01C0,0908,0000,1376  ;3590  Q_RC[1]     ; Get CODE_VECTOR
U 1376, 081D,2020,0180,0AB0,0000,1377  ;3591  Q_Q.XOR.D,R[6]_ALU  ; RCVD_VECTOR = RCVD_VECTOR .XOR.
          ;3592       ; ... CODE_VECTOR
U 1377, 0003,003C,0180,0AB8,0000,1378  ;3593  R[7]_0     ; Initialize ONE_COUNTER to 0
U 1378, 0018,0038,0580,0AD0,0000,1379  ;3594  R[0A]_K[.1]   ; Set Bit 0 of SYMBOL_POINTER
          ;3595 DECODE.ZERO.ORDR.LOOP:
U 1379, 0800,003C,0180,0A50,0000,137A  ;3596  D_R[0A]     ; Get SYMBOL_POINTER
          ;3597  ALU_D[AND]R[6],    ; Is SYMBOL_POINTER .AND. RCVD_VECTOR
U 137A, 000D,0034,0180,0A30,0010,137B  ;3598   CLK.UBCC  ; ...equal to 0?
          ;3599  Z?,
U 137B, 0800,013C,0180,0A38,0000,1188  ;3600   D_R[7]    ; Get ONE_COUNTER
          ;3601 =0 J/DECODE.ZERO.ORDR.BITNZ, ; Bit was NOT 0 in RCVD_VECTOR
U 1188, 0019,0014,0580,0AB8,0000,137C  ;3602   R[7]_D+K[.1]  ; Increment ONE_COUNTER
          ;3603  J/DECODE.ZERO.ORDR.BITZ, ; Bit was equal to 0
U 1189, 0800,003C,0180,0A50,0000,137F  ;3604   D_R[0A]    ; Get SYMBOL_POINTER
          ;3605 DECODE.ZERO.ORDR.BITNZ:
U 137C, 0800,003C,0180,0A30,0000,137D  ;3606  D_R[6]     ; Get RCVD_VECTOR
          ;3607  ALU_D[ANDNOT]R[0A],D_ALU, ; RCVD_VECTOR = RCVD_VECTOR .ANDNOT.
U 137D, 080D,0024,0180,0A50,0010,137E  ;3608   CLK.UBCC  ; ...SYMBOL_POINTER (i.e., clear bit
          ;3609       ; ...tested and see if the overall
          ;3610       ; ...result is 0)
U 137E, 0000,013C,0180,0800,0000,118A  ;3611  Z?
          ;3612 =0 J/DECODE.ZERO.ORDR.BITZ, ; RCVD_VECTOR is NOT zero so
U 118A, 0800,003C,0180,0A50,0000,137F  ;3613   D_R[0A]    ; ...continue decoding all zero
          ;3614       ; ...order bits
          ;3615  J/DECODE.ZERO.ORDR.DONE, ; RCVD_VECTOR is 0 so we finished
U 118B, 0800,003C,0180,0A38,0000,1381  ;3616   D_R[7]    ; ...decoding all zero order bits.
          ;3617 DECODE.ZERO.ORDR.BITZ:
          ;3618  R[0A]_D.LEFT,SI/ZERO,  ; Arithmetic Shift Left SYMBOL_POINTER.
U 137F, 0021,003C,0180,0AD0,0010,1380  ;3619   CLK.UBCC  ; ...and see if it is 0
U 1380, 0000,013C,0180,0800,0000,118C  ;3620  Z?
U 118C, 0000,003C,0180,0800,0000,1379  ;3621 =0 J/DECODE.ZERO.ORDR.LOOP  ; ...Continue decoding
U 118D, 0800,003C,0180,0A38,0000,1381  ;3622  D_R[7]     ; Get ONE_COUNTER
          ;3623 DECODE.ZERO.ORDR.DONE:
          ;3624  ALU_D-K[.10],    ; Is ONE_COUNTER GREATER than or EQUAL
U 1381, 0019,0000,6580,0800,0010,1382  ;3625   CLK.UBCC  ; ...to d16.
          ;3626  ALU.N?,
U 1382, 0810,1B38,0180,0910,0000,1097  ;3627   D_RC[2]    ; Get INFO_SYMBOLS
U 1097, 0019,0030,0580,0990,0000,109F  ;3628 =0111 RC[2]_D.OR.K[.1]  ; Set Bit 0 of INFO_SYMBOLS
U 109F, 0000,003C,0180,0800,0000,1383  ;3629  NOP
          ;3630 ;
          ;3631 ;-
          ;3632 ;
```

```
            ;3633 ; This is where the actual decoding of the received Reed-Muller
            ;3634 ; code (RCVD_VECTOR) ends.  The decoded result is left in
            ;3635 ; RC[2]  (INFO_SYMBOLS) which will be now compared against
            ;3636 ; the expected value which is in RC[4].
            ;3637 ;
            ;3638 ;++
            ;3639 ;
U 1383, 0810,0038,0180,0910,0000,1384  ;3640  D_RC[2]     ; Get INFO_SYMBOLS
            ;3641  ALU_D[XOR]RC[4],   ; See if INFO_SYMBOLS is equal to
U 1384, 0011,0020,0180,0920,0010,1385  ;3642   CLK.UBCC   ; ...the expected data
U 1385, 0000,013C,0180,0800,0000,118E  ;3643  Z?
            ;3644 =0 J/DECODE.ERROR,   ; Data Returned is not equal to
U 118E, 0000,003C,0180,0800,1480,1386  ;3645  SC_STATE    ; ...the expected.  Get number of
            ;3646       ; ...errors
U 118F, 0000,003E,0180,0800,0000,0003  ;3647  RETURN3     ; OK. No error in data so do a
            ;3648       ; ...RETURN3
            ;3649 DECODE.ERROR:
U 1386, 0000,003C,E980,0800,0084,9387  ;3650  SC_SC+K[.24]    ; Get number of errors and transform it
            ;3651       ; ...in an ID Register address
            ;3652       ; ...starting with ID register 25
U 1387, 0000,003C,0580,0800,0084,9388  ;3653  SC_SC+K[.1]    ; SC Reg holds a pointer to ID regs 25
            ;3654       ; ...through 2X depending on how many
            ;3655       ; ...errors occurred.
U 1388, 0810,0038,0180,0910,0000,1389  ;3656  D_RC[2]     ; Get Received DECODED data
U 1389, 0000,003C,0180,3400,0000,138A  ;3657  ID(SC)_D    ; Save it in ID Reg pointed by SC
            ;3658  D_RC[4],    ; Get Expected Data
U 138A, 0810,0038,0580,0920,0084,938B  ;3659   SC_SC+K[.1]  ; Increment pointer to ID Regs
U 138B, 0000,003C,0180,3400,0000,138C  ;3660  ID(SC)_D    ; Save Expected data in ID reg
U 138C, 0000,003C,0980,0800,1404,938D  ;3661  STATE_STATE+K[.2]  ; Increment the error counter.
U 138D, 0000,003C,0180,0800,1480,138E  ;3662  SC_STATE    ; Get the number of errors
U 138E, 0818,0038,1D80,0800,0000,138F  ;3663  D_SC     ; ...in D reg
            ;3664  ALU_D.XOR.K[.C],   ; Is the error counter equal to
U 138F, 0019,0020,8580,0800,0010,1390  ;3665   CLK.UBCC   ; ...x0C? (i.e., maximum number of
            ;3666       ; ...errors has been reached)
U 1390, 0000,013C,0180,0800,0000,1190  ;3667  Z?
U 1190, 0000,003E,0180,0800,0000,0002  ;3668 =0 RETURN2     ; Maximum number of errors has NOT been
            ;3669       ; ...reached yet
U 1191, 0000,003E,0180,0800,0000,0001  ;3670  RETURN1     ; Maximum number of errors!!
            ;3671
            ;3672
            ;3673
            ;3674
```

```
;3675  .PAGE
;3676  .TOC "   FIND CONTROLLERS"
;3677 ;*************************************************************
;3678 ;++
;3679 ;
;3680 ; Functional Description:
;3681 ; ---------------------
;3682 ;
;3683 ;
;3684 ;      This subroutine is called by tests that need to know how
;3685 ; many  controllers are available on the MS780-E/H currently under
;3686 ; test.  It makes use of subroutine SELECT.CNTRLR to put the
;3687 ; controllers in different Interleave Modes.
;3688 ; The subroutine returns control to the calling routine by either
;3689 ; one of:  Return1, Return2, Return3 or Return4.  It is up to the
;3690 ; test to decode this information as necessary.
;3691 ; Since some Tests require this feature, the subroutine also finds
;3692 ; the memory size available on each controller and leaves it in
;3693 ; registers R[0] and R[1].
;3694 ;
;3695 ;
;3696 ; Calling Constraint: =000
;3697 ; ------------------
;3698 ;
;3699 ;
;3700 ; Inputs:
;3701 ; ------
;3702 ;
;3703 ; RC[0E] MUST HOLD THE I/O BASES ADDRESS OF THE MS780-E/H
;3704 ; UNDER TEST
;3705 ;
;3706 ; Outputs:
;3707 ; -------
;3708 ;
;3709 ; - Return1, Return2, Return3 or Return4 depending on
;3710 ;   how many Controllers were found
;3711 ;
;3712 ; - R[0] Size of Memory on the Lower Controller
;3713 ;   R[1] Size of Memory on the Upper Controller
;3714 ;
;3715 ; Side Effects:
;3716 ; ------------
;3717 ;
;3718 ; The contents of the following registers will be lost:
;3719 ;
;3720 ; D, Q, SC, R[0], R[1]
;3721 ;
;3722 ;--
;3723 ;*************************************************************
;3724 =00
;3725 FIND.CNTRLRS:
;3726  D_K[ZERO],   ; Attempt to select the Lower
U 1030, 0818,0039,1980,0800,0000,1263  ;3727   CALL[SELECT.CNTRLR] ; ...Controller
U 1031, 0000,003C,0180,0800,0000,1038  ;3728  J/LOW.MIS.X   ; Lower Controller Misconfigured
U 1032, 0000,003D,0180,0800,0000,12D7  ;3729  CALL[GET.MEMORY.SIZE]  ; Lower Controller has been  configured
```

```
            ;3730       ; ...find its memory size
U 1033, 0001,003C,0180,0A80,0000,1034  ;3731  R[0]_D     ; Save Memory size in R[0]
            ;3732 =00 D_K[.2],    ; Attempt to select the Upper
U 1034, 0818,0039,0980,0800,0000,1263  ;3733    CALL[SELECT.CNTRLR] ; ...Controller
U 1035, 0000,003E,0180,0800,0000,0002  ;3734  RETURN2     ; Upper Controller Misconfigured.
            ;3735       ; ...Let caller know that ONLY the
            ;3736       ; ...Lower Controller was found
U 1036, 0000,003D,0180,0800,0000,12D7  ;3737  CALL[GET.MEMORY.SIZE]  ; Upper Controller was Configured so
            ;3738       ; ...find its memory size
            ;3739  R[1]_D,     ; Save Memory Size in R[1]
U 1037, 0001,003E,0180,0A88,0000,0004  ;3740   RETURN4    ; Let Caller know that BOTH Controllers
            ;3741       ; ...were found
            ;3742 =00
            ;3743 LOW.MIS.X:
            ;3744  D_K[.2],    ; Lower Controller was not found,
U 1038, 0818,0039,0980,0800,0000,1263  ;3745   CALL[SELECT.CNTRLR] ; ...so attempt to Configure the
            ;3746       ; ...Upper Controller
U 1039, 0000,003E,0180,0800,0000,0001  ;3747  RETURN1     ; Tell Caller that NONE of the
            ;3748       ; ...Controllers were found
U 103A, 0000,003D,0180,0800,0000,12D7  ;3749  CALL[GET.MEMORY.SIZE]  ; Upper Controller was found,
            ;3750       ; ...go get its memory size
            ;3751  R[1]_D,     ; Save the memory size in R[1]
U 103B, 0001,003E,0180,0A88,0000,0003  ;3752   RETURN3    ; Let Caller know that ONLY the
            ;3753       ; ...Upper Controller was found
            ;3754
            ;3755
```

```
;3756  .PAGE "TEST 1B7 CONTROLLER SELECTION"
;3757 ;*************************************************************
;3758 ;+++
;3759 ;
;3760 ; Functional Description:
;3761 ; ---------------------
;3762 ;
;3763 ; This test will check the controller selection logic.  It will place the
;3764 ; controllers in all possibile interleave modes and will write unique
;3765 ; data in location 0.
;3766 ;
;3767 ; This test, as well as some of the folowing tests, make use of Reed-
;3768 ; Muller codes which allow for a few bits in error in the data stored on
;3769 ; the array.  These codes are being generated by subroutine ENCODE and
;3770 ; decoded by the subroutine DECODE.
;3771 ;
;3772 ; In any of the following tests up to 6 addressing errors are allowed, in
;3773 ; which case the error data will be retreived from ID registers 24
;3774 ; through 30 and will be placed in RC registers x0C through x00.  This
;3775 ; will be accomplished by a subroutine ERDAT which is defined in the
;3776 ; current micro2.
;3777 ;
;3778 ;   [Subtest 1:]
;3779 ;
;3780 ; This subtest first places the lower controller in the Non-Interleaved
;3781 ; mode and writes an encoded test pattern to location 0.  The controller
;3782 ; is then placed in the Externally Interleaved Mode and the integrity of
;3783 ; location 0 ( or 8 depending on the TR level) is checked.  Still in
;3784 ; Externally Interleaved Mode a different encoded pattern is written to
;3785 ; location 0.  The controller is then put back in Non-Interleaved Mode
;3786 ; and the integrity of location 0 is checked.  It is expected that no
;3787 ; micro traps or Interrupts will occur and the data is written correctly
;3788 ; on the array.  The only Time-Out is expected when operating in
;3789 ; Externally Interleaved Mode and a non existent location is accessed.
;3790 ;
;3791 ;   [Subtest 2:]
;3792 ;
;3793 ; This subtest is basically the same as the previous one, except that the
;3794 ; Upper Controller is tested.  Note:  If one of the Controllers is
;3795 ; missing then only the test partaining to the specific controller will
;3796 ; be executed and the other will be skipped.
;3797 ;
;3798 ;   [Subtest 3:]
;3799 ;
;3800 ; This subtest places the controllers in the internally interleaved mode.
;3801 ; Different encoded data patterns are written to locations 0 and 8.  The
;3802 ; controllers are then placed in non-interleave mode and locations 0 on
;3803 ; both controllers are read and checked for data integrity.  Still in
;3804 ; non-interleaved mode different encoded data patterns are written to
;3805 ; location 0 of each controller.  Placing the controllers in internally
;3806 ; interleaved mode the data from locations 0 and 8 is check for
;3807 ; integrity.
;3808 ;
;3809 ;
;3810 ;
```

```
            ;3811 ; Error Logout:
            ;3812 ; ------------
            ;3813 ;
            ;3814 ; See individual error reports.
            ;3815 ;
            ;3816 ;
            ;3817 ;=============================================================
            ;3818 ;
            ;3819 ; Register Map:
            ;3820 ; ------------
            ;3821 ;
            ;3822 ;     RA,RB     Description:
            ;3823 ;     -----     -----------
            ;3824 ;
            ;3825 ; Not Used by main-line code (used in Subroutines).
            ;3826 ;
            ;3827 ;      RC       Description:
            ;3828 ;      --       -----------
            ;3829 ;
            ;3830 ;       E I/O Base Address of MS780-E currently under test
            ;3831 ;
            ;3832 ;       D Memory Test Address
            ;3833 ;
            ;3834 ;       C Controller Present Flag  1 = Upper &  Lower Present
            ;3835 ;      0 = Upper or Lower Presnet
            ;3836 ;
            ;3837 ;--
            ;3838 ;*************************************************************
            ;3839 =0
U 1192, 0000,003D,0180,0800,0000,109B  ;3840 T.29: NEWTST   ; Signify start of new test
U 1193, 0000,003C,0180,0800,0000,1040  ;3841  NOP
U 1040, 0000,003D,0180,0800,0000,1094  ;3842 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1041, 0000,003C,0180,0800,0000,113B  ;3843  J/T.29.EXIT  ; No MS780-E's on the system, exit this test
U 1042, 0000,003C,0180,0800,0000,1010  ;3844  nop
            ;3845 =
            ;3846  ;
            ;3847  ; This subroutine (FIND.CNTRLRS) is used to determine the configuration
            ;3848  ; of controller(s) on the MS780-E that is currently being tested.
            ;3849  ; There are 4 cases that are a result of this subroutine call.  These
            ;3850  ; cases are:
            ;3851  ;
            ;3852  ; CASE 1 No controllers found on this MS780-E
            ;3853  ; CASE 2 Only the lower controller was found on this MS780-E
            ;3854  ; CASE 3 Only the upper controller was found on this MS780-E
            ;3855  ; CASE 4 Both the upper and lower controller were found on this
            ;3856  ;  MS780-E
            ;3857  ;
            ;3858  ; This subroutine will also return the memory size for each controller.
            ;3859  ; The memory size for the lower controller is returned in R[0] and the
            ;3860  ; memory size for the upper controller is returned in R[1].
            ;3861  ;
            ;3862 =000
            ;3863 RESTART.TEST.29:
U 1010, 0000,003D,0180,0800,0000,1030  ;3864  CALL[FIND.CNTRLRS] ; Determine what controllers are on this MS780
U 1011, 0000,003C,0180,0800,0000,1391  ;3865  J/T.29.NO.CTRL  ; CASE 1 - Return here if there were no
```

```
            ;3866     ; controllers found.  Call an error.
U 1012, 0000,003C,0180,0800,0000,1392  ;3867  J/T.29.LOW.ONLY  ; CASE 2 - Return here if only the lower
            ;3868     ; controller was found.
U 1013, 0000,003C,0180,0800,0000,1194  ;3869  J/T.29.UP.ONLY  ; CASE 3 - Return here if only the upper
            ;3870     ; controller was found.
U 1014, 0000,003C,0180,0800,0000,1394  ;3871  J/T.29.BOTH.CTRL ; CASE 4 - Return here if both controllers
            ;3872     ; were found.
            ;3873 =
```

```
            ;3874  .PAGE
            ;3875 ;
            ;3876 ;////////////////////////////////////////////////////////////////////////////
            ;3877 ;
            ;3878 ; This error call is executed when there were no controllers found from the
            ;3879 ; call to FIND.CNTRLRS.  The test is NOT restartable after this error.
            ;3880 ;
            ;3881 T.29.NO.CTRL:
U 1391. 0000,003D,0580,0800,0084,703C  ;3882  ERROR1[.1]
            ;3883 ;
            ;3884 ;////////////////////////////////////////////////////////////////////////////
            ;3885 ;
            ;3886 ; ERROR LOGOUT:
            ;3887 ;
            ;3888 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3889 ;
            ;3890 ;////////////////////////////////////////////////////////////////////////////
            ;3891 ;
            ;3892  ;
            ;3893  ; Come here when there is only a lower controller found on the MS780-E
            ;3894  ; currently under test.  Increment the Subtest counter as we go from
            ;3895  ; here directly into Subtest 2.  R0 is used as a flag to signify that
            ;3896  ; there were Both ( R0 = 1 ) or only One ( R0 = 0 ) controller(s)
            ;3897  ; found.
            ;3898  ;
            ;3899 T.29.LOW.ONLY:
            ;3900  RC[0C]_0,  ; Clear Flag.  This will be used in Subtests 2
            ;3901      ; and 3 to see whether both controllers present
U 1392. 0003,003C,0180,09E0,0000,1196  ;3902   J/T.29.S1 ; Jump to Subtest 1
            ;3903  ;
            ;3904  ; Come here when there is only an upper controller found on the MS780-E
            ;3905  ; currently under test.  Increment the Subtest number twice as we go
            ;3906  ; directly from here to Subtest 3.
            ;3907  ;
            ;3908 =0
            ;3909 T.29.UP.ONLY:
U 1194. 0000,003D,0180,0800,0000,1261  ;3910  SUBTEST   ; Increment the subtest counter twice
            ;3911      ; because we are going to execute
U 1195. 0000,003C,1980,0800,1404,7393  ;3912  STATE_K[ZERO]  ; Clear the State register used by decode/erdat
            ;3913  RC[0C]_0,  ; Clear Flag.  This will be used in Subtests 2
            ;3914      ; and 3 to see whether both controllers present
U 1393. 0003,003C,0180,09E0,0000,11D2  ;3915   J/T.29.S2 ; Jump to Subtest 2
            ;3916  ;
            ;3917  ; Come here when both controllers were found on the MS780-E currently
            ;3918  ; under test.
            ;3919  ;
            ;3920 T.29.BOTH.CTRL:
U 1394. 0018,0038,0580,09E0,0000,1196  ;3921  RC[0C]_K[.1]  ; Set flag to signify that both controllers
            ;3922      ; were found.  Fall through to Subtest 1.
            ;3923 ;
            ;3924 ;============================================================================
            ;3925 ;
            ;3926 ; Subtest 1
            ;3927 ;
            ;3928 =0
```

```
            ;3929 T.29.S1:
U 1196. 0000.003D.0180.0800.0000.1261  ;3930  SUBTEST    ; Increment Subtest counter
U 1197. 0000.003C.1980.0800.1404.7395  ;3931  STATE_K[ZERO]  ; Clear the State register used by decode/erdat
U 1395. 0003.003C.0180.09E8.0000.1396  ;3932  RC[0D]_0  ; Load RC[0D] with test address
U 1396. 0018.0038.0580.09A0.0000.1198  ;3933  RC[04]_K[.1]  ; Set pattern count to 1
U 1198. 0000.003D.0180.0800.0000.12F3  ;3934 =0 CALL[ENCODE]  ; Encode the pattern
U 1199. 0000.003C.0180.0800.0000.119A  ;3935  NOP
U 119A. 0000.003D.0180.0800.0000.1259  ;3936 =0 ERLOOP    ; Loop on error from here
U 119B. 0000.003C.0180.0800.0000.119C  ;3937  NOP
U 119C. 0000.003D.0180.0800.0000.107C  ;3938 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 119D. 0000.003C.0180.0800.0000.119E  ;3939  NOP
U 119E. 0818.0039.0D80.0800.0000.12E7  ;3940 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 119F. 0000.003C.0180.0800.0000.1397  ;3941  NOP
            ;3942  ;
            ;3943  ; Try to put the memory into Non-Interleave mode
            ;3944  ;
U 1397. 0818.0038.1980.0800.0000.1048  ;3945  D_K[ZERO]  ; Move value to specify Non-Interleave Lower
            ;3946      ; to the D register for the next call
U 1048. 0000.003D.0180.0800.0000.1263  ;3947 =00 CALL[SELECT.CNTRLR] ; Select Non-Interleaved mode
```

```
            ;3948  .PAGE
            ;3949 ;
            ;3950 ;////////////////////////////////////////////////////////////////////////////////
            ;3951 ;
            ;3952 ; Got a misconfiguration error when selecting Non-Interleaved mode.  Call an
            ;3953 ; error
            ;3954 ;
            ;3955 T.29.S1.CONFIG.ERR:
U 1049, 0000,003D,0D80,0800,0084,703C  ;3956  ERROR1[.3]
            ;3957 ;
            ;3958 ;////////////////////////////////////////////////////////////////////////////////
            ;3959 ;
            ;3960 ; ERROR LOGOUT:
            ;3961 ;
            ;3962 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3963 ;  Memory Test Address
            ;3964 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;3965 ;       0 = Upper or Lower Present
            ;3966 ;
            ;3967 ;////////////////////////////////////////////////////////////////////////////////
            ;3968 ;
            ;3969  ;
            ;3970  ; No misconfiguration error when putting memory in Non-Interleave mode.
            ;3971  ; Now lets do a write to memory with the encoded pattern.
            ;3972  ;
            ;3973 T.29.S1.NO.CONFIG.ERR:
U 104A, 0010,0038,0180,0968,0200,104B  ;3974  VA_RC[0D]  ; Load VA with test address
U 104B, 0810,0038,0180,0908,0000,1398  ;3975  D_RC[01]  ; Load D with encoded vector
U 1398, 0000,003C,0180,A800,0000,1050  ;3976  CACHE.P_D[LONG]  ; Write the coded vector to the location
U 1050, 0000,003D,0180,0800,0000,1280  ;3977 =00 CHECK.INTERRUPT  ; Were there any unexpected interrupts ?
U 1051, 0000,003C,0180,0800,0000,1070  ;3978  J/T.29.S1.CHK.TR ; No, continue with test
```

```
            ;3979   .PAGE
            ;3980  ;
            ;3981  ;////////////////////////////////////////////////////////////////////////////////
            ;3982  ;
            ;3983  ; Got an unexpected interrupt when writing the encoded value out to memory.
            ;3984  ; Call an error
            ;3985  ;
U 1052. 0000.003D.F580.0800.0084.703C   ;3986   ERROR1[.A]
            ;3987  ;
            ;3988  ;////////////////////////////////////////////////////////////////////////////////
            ;3989  ;
            ;3990  ; ERROR LOGOUT:
            ;3991  ;
            ;3992  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3993  ;  Memory Test Address
            ;3994  ;  Controller Present Flag  1 = Upper &  Lower Present
            ;3995  ;       0 = Upper or Lower Present
            ;3996  ;  Unused
            ;3997  ;  Unused
            ;3998  ;  Unused
            ;3999  ;  ID[vector] Register
            ;4000  ;  ID[sbi.err] Register
            ;4001  ;  ID[fault] Register
            ;4002  ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4003  ;       0 = No Interrupt Occurred
            ;4004  ;
            ;4005  ;////////////////////////////////////////////////////////////////////////////////
            ;4006  ;
            ;4007  =
            ;4008  =00
            ;4009  T.29.S1.CHK.TR:
U 1070. 0000.003D.0180.0800.0000.12ED   ;4010   CALL[ODD.EVEN.TR] ; Is this memory at odd or even TR ?
            ;4011    RC[0D]_K[.8],   ; Even TR
U 1071. 0018.0038.0180.09E8.0000.11A0   ;4012   J/T.29.S1.SEC.2
U 1072. 0018.0038.1980.09E8.0000.11A0   ;4013   RC[0D]_K[ZERO]   ; Odd TR
            ;4014  =
            ;4015  =0
            ;4016  T.29.S1.SEC.2:
U 11A0. 0000.003D.0180.0800.0000.1259   ;4017   ERLOOP    ; Loop on error from here
U 11A1. 0000.003C.0180.0800.0000.11A2   ;4018   NOP
U 11A2. 0000.003D.0180.0800.0000.107C   ;4019  =0 CALL[SBI.INIT]   ; Initialize the SBI
U 11A3. 0000.003C.0180.0800.0000.11A4   ;4020   NOP
U 11A4. 0818.0039.0D80.0800.0000.12E7   ;4021  =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 11A5. 0000.003C.0180.0800.0000.1399   ;4022   NOP
U 1399. 0818.0038.0580.0800.0000.1074   ;4023   D_K[.1]    ; Load D with 1 for Externally Interleaved
U 1074. 0000.003D.0180.0800.0000.1263   ;4024  =00 CALL[SELECT.CNTRLR] ; Select Externally Interleaved mode
```

```
            ;4025  .PAGE
            ;4026 ;
            ;4027 ;////////////////////////////////////////////////////////////////////////////////
            ;4028 ;
            ;4029 ; Got a misconfiguration error when selecting Non-Interleave mode.  Call an
            ;4030 ; error
            ;4031 ;
            ;4032 T.29.S1.CONFIG.ERR.2:
U 1075, 0000.003D.0D80.0800.0084.703C  ;4033  ERROR1[.3]
U 1076, 0000.003C.0180.0800.0000.11A6  ;4034  NOP
            ;4035 =
            ;4036 ;
            ;4037 ;////////////////////////////////////////////////////////////////////////////////
            ;4038 ;
            ;4039 ; ERROR LOGOUT:
            ;4040 ;
            ;4041 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4042 ;  Memory Test Address  0 = Odd TR  0 = Even TR
            ;4043 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4044 ;       0 = Upper or Lower Present
            ;4045 ;
            ;4046 ;////////////////////////////////////////////////////////////////////////////////
            ;4047 ;
            ;4048 =0
            ;4049 T.29.S1.CONT.SEC.2:
U 11A6, 0000.003D.8980.0800.0084.7293  ;4050  SET.TRAP.MASK[.D] ; Enable Time out micro traps
U 11A7, 0010.0038.0180.0968.0200.139A  ;4051  VA_RC[0D]  ; Point VA to the test location
U 139A, 0000.003C.0180.C800.0000.139B  ;4052  D[LONG]_CACHE.P  ; Read the test location
U 139B, 0001.003C.0180.0980.0000.1078  ;4053  RC[00]_D  ; Save the read data in RC[00]
U 1078, 0000.003D.0180.0800.0000.128D  ;4054 =00 CHECK.UTRAP  ; Check for any micro traps
U 1079, 0000.003C.0180.0800.0000.1080  ;4055  J/T.29.S1.CHK.INT.1 ; No, go check for any unexpected interrupts
```

```
            ;4056  .PAGE
            ;4057 ;
            ;4058 ;////////////////////////////////////////////////////////////////////////////////
            ;4059 ;
            ;4060 ; Got a time out micro trap when reading the data from the test location.
            ;4061 ; Call an error.
            ;4062 ;
U 107A, 0000.003D.0180.0800.0084.7044  ;4063  ERROR2[.8]
U 107B, 0000.003C.0180.0800.0000.1080  ;4064  NOP
            ;4065 ;
            ;4066 ;////////////////////////////////////////////////////////////////////////////////
            ;4067 ;
            ;4068 ; ERROR LOGOUT:
            ;4069 ;
            ;4070 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4071 ;  Memory Test Address
            ;4072 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4073 ;       0 = Upper or Lower Present
            ;4074 ;  Unused
            ;4075 ;  Unused
            ;4076 ;  Unused
            ;4077 ;  Expected Micro Trap Flags
            ;4078 ;  Received Micro Trap Flags
            ;4079 ;
            ;4080 ;////////////////////////////////////////////////////////////////////////////////
            ;4081 ;
            ;4082 =00
            ;4083 T.29.S1.CHK.INT.1:
U 1080, 0000.003D.0180.0800.0000.1280  ;4084  CHECK.INTERRUPT  ; Any unexpected Interrupts ?
U 1081, 0000.003C.0180.0800.0000.1084  ;4085  J/T.29.S1.DO.DECODE ; No, no go decode the data
```

```
            ;4086  .PAGE
            ;4087 ;
            ;4088 ;////////////////////////////////////////////////////////////////////////////////
            ;4089 ;
            ;4090 ; Got an unexpected interrupt when reading the data from the test location.
            ;4091 ; Call an error.
            ;4092 ;
U 1082, 0000,003D,F580,0800,0084,703C  ;4093  ERROR1[.A]
U 1083, 0000,003C,0180,0800,0000,1084  ;4094  NOP
            ;4095 ;
            ;4096 ;////////////////////////////////////////////////////////////////////////////////
            ;4097 ;
            ;4098 ; ERROR LOGOUT:
            ;4099 ;
            ;4100 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4101 ;  Memory Test Address
            ;4102 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4103 ;       0 = Upper or Lower Present
            ;4104 ;  Unused
            ;4105 ;  Unused
            ;4106 ;  Unused
            ;4107 ;  ID[vector] Register
            ;4108 ;  ID[sbi.err] Register
            ;4109 ;  ID[fault] Register
            ;4110 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4111 ;       0 = No Interrupt Occurred
            ;4112 ;
            ;4113 ;////////////////////////////////////////////////////////////////////////////////
            ;4114 ;
            ;4115 =00
            ;4116 T.29.S1.DO.DECODE:
U 1084, 0000,003D,0180,0800,0000,1317  ;4117  CALL[DECODE]  ; Check the data returned
U 1085, 0000,003C,0180,0800,0000,1086  ;4118  NOP    ; Return1 for maximum errors
U 1086, 0000,003C,0180,0800,0000,11A8  ;4119  J/T.29.CALL.ERDAT.1 ; Return2 for errors but not maximum
U 1087, 0000,003C,0180,0800,0000,11AC  ;4120  J/T.29.S1.READ.NXM ; Data returned was good, try reading a
            ;4121      ; Non Existent Memory location
```

```
            ;4122  .PAGE
            ;4123 ;
            ;4124 ;////////////////////////////////////////////////////////////////////////////////
            ;4125 ;
            ;4126 ; Data that was decoded from the read of the test location was bad.  Call an
            ;4127 ; error.
            ;4128 ;
            ;4129 =0
            ;4130 T.29.CALL.ERDAT.1:
U 11A8, 0000.003D.0180.0800.0000.12DE  ;4131  CALL[ERDAT]  ; Set up the registers with the error data
U 11A9, 0000.003C.0180.0800.0000.11AA  ;4132  NOP
U 11AA, 0000.003D.6180.0800.0084.7044  ;4133 =0 ERROR2[.F]
U 11AB, 0000.003C.0180.0800.0000.11AC  ;4134  NOP
            ;4135 ;
            ;4136 ;////////////////////////////////////////////////////////////////////////////////
            ;4137 ;
            ;4138 ; ERROR LOGOUT:
            ;4139 ;
            ;4140 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4141 ;  Memory Test Address
            ;4142 ;  Received Memory Data
            ;4143 ;  Expected Memory Data
            ;4144 ;  Received Memory Data
            ;4145 ;  Expected Memory Data
            ;4146 ;  Received Memory Data
            ;4147 ;  Expected Memory Data
            ;4148 ;  Received Memory Data
            ;4149 ;  Expected Memory Data
            ;4150 ;  Received Memory Data
            ;4151 ;  Expected Memory Data
            ;4152 ;  Received Memory Data
            ;4153 ;  Expected Memory Data
            ;4154 ;  FFFFFFFF
            ;4155 ;
            ;4156 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4157 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4158 ;        be output on the error then the last longword wrote will be the
            ;4159 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4160 ;
            ;4161 ;////////////////////////////////////////////////////////////////////////////////
            ;4162 ;
            ;4163 =0
            ;4164 T.29.S1.READ.NXM:
U 11AC, 0000.003D.0180.0800.0000.1259  ;4165  ERLOOP   ; Loop on error from here
U 11AD, 0010.0038.0180.0968.0200.11AE  ;4166  VA_ALU,ALU_RC[0D] ; Point to a non-existent memory location
            ;4167 =0 CALL[SBI.INIT],  ; Initialize the SBI
U 11AE, 0000.003D.0180.0803.0000.107C  ;4168   VA_VA+4  ; VA should be pointing to location
U 11AF, 0000.003C.0180.0803.0000.11B0  ;4169  VA_VA+4   ; ...8 or C
U 11B0, 0000.003D.8980.0800.0084.7293  ;4170 =0 SET.TRAP.MASK[.D] ; Enable time out micro traps
U 11B1, 0000.003C.0180.C800.0000.1088  ;4171  D[LONG]_CACHE.P  ; Try reading the nonexistent memory location
U 1088, 0000.003D.0180.0800.0000.128D  ;4172 =00 CHECK.UTRAP  ; Was there a time out micro trap
U 1089, 0000.003C.0180.0800.0000.11B2  ;4173  J/T.29.S1.NOTMO.ERR ; No, there should have been, call an error
U 108A, 0000.003C.0180.0800.0000.11B3  ;4174  J/T.29.S1.DO.ENCODE ; Yes, continue with test
            ;4175 =
```

```
            ;4176  .PAGE
            ;4177 ;
            ;4178 ;////////////////////////////////////////////////////////////////////////////////
            ;4179 ;
            ;4180 ; Didn't get a time out micro trap when reading a non-existent memory location.
            ;4181 ; Call an error.  Should have got time out.
            ;4182 ;
            ;4183 =0
            ;4184 T.29.S1.NOTMO.ERR:
U 11B2, 0000,003D,0180,0800,0084,7044  ;4185  ERROR2[.8]
            ;4186 ;
            ;4187 ;////////////////////////////////////////////////////////////////////////////////
            ;4188 ;
            ;4189 ; ERROR LOGOUT:
            ;4190 ;
            ;4191 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4192 ;  Memory Test Address
            ;4193 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4194 ;       0 = Upper or Lower Present
            ;4195 ;  Unused
            ;4196 ;  Unused
            ;4197 ;  Unused
            ;4198 ;  Expected Micro Trap Flags
            ;4199 ;  Received Micro Trap Flags
            ;4200 ;
            ;4201 ;////////////////////////////////////////////////////////////////////////////////
            ;4202 ;
            ;4203 T.29.S1.DO.ENCODE:
U 11B3, 0810,0038,0180,0920,0000,139C  ;4204  D_RC[04]  ; Move pattern select (RC[04]) to the D reg
U 139C, 0019,0014,0980,09A0,0000,11B4  ;4205  RC[04]_D+K[.2]  ; Add 2 to select new pattern
U 11B4, 0000,003D,0180,0800,0000,12F3  ;4206 =0 CALL[ENCODE]  ; Encode the new pattern
U 11B5, 0000,003C,0180,0800,0000,11B6  ;4207  NOP
U 11B6, 0000,003D,0180,0800,0000,1259  ;4208 =0 ERLOOP   ; Loop on error from here
U 11B7, 0000,003C,0180,0800,0000,11B8  ;4209  NOP
U 11B8, 0000,003D,0180,0800,0000,107C  ;4210 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11B9, 0000,003C,0180,0800,0000,11BA  ;4211  NOP
U 11BA, 0818,0039,0D80,0800,0000,12E7  ;4212 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 11BB, 0010,0038,0180,0968,0200,139D  ;4213  VA_RC[0D]  ; Load VA with 0 for memory location 0
U 139D, 0810,0038,0180,0908,0000,139E  ;4214  D_RC[01]  ; Move pattern returned from Encode to D reg
U 139E, 0000,003C,0180,A800,0000,1090  ;4215  CACHE.P_D[LONG]  ; Write the pattern to memory
U 1090, 0000,003D,0180,0800,0000,1280  ;4216 =00 CHECK.INTERRUPT  ; Any unexpected interrupts on the write ?
U 1091, 0000,003C,0180,0800,0000,1093  ;4217  J/T.29.S1.CHK.READ ; No, get ready to read the data back
```

```
            ;4218  .PAGE
            ;4219 ;
            ;4220 ;////////////////////////////////////////////////////////////////////////////////
            ;4221 ;
            ;4222 ; Got an unexpected interrupt when writing the data to the test location.
            ;4223 ; Call an error.
            ;4224 ;
U 1092, 0000,003D,F580,0800,0084,703C  ;4225  ERROR1[.A]
            ;4226 ;
            ;4227 ;////////////////////////////////////////////////////////////////////////////////
            ;4228 ;
            ;4229 ; ERROR LOGOUT:
            ;4230 ;
            ;4231 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4232 ;  Memory Test Address
            ;4233 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4234 ;       0 = Upper or Lower Present
            ;4235 ;  Unused
            ;4236 ;  Unused
            ;4237 ;  Unused
            ;4238 ;  ID[vector] Register
            ;4239 ;  ID[sbi.err] Register
            ;4240 ;  ID[fault] Register
            ;4241 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4242 ;       0 = No Interrupt Occurred
            ;4243 ;
            ;4244 ;////////////////////////////////////////////////////////////////////////////////
            ;4245 ;
            ;4246 T.29.S1.CHK.READ:
U 1093, 0003,003C,0180,09E8,0000,11BC  ;4247  RC[0D] 0  ; Set RC[0D] to 0, location to be accessed
U 11BC, 0000,003D,0180,0800,C000,1259  ;4248 =0 ERLOOP   ; Loop on error from here
U 11BD, 0000,003C,0180,0800,0000,11BE  ;4249  NOP
U 11BE, 0000,003D,0180,0800,0000,107C  ;4250 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11BF, 0000,003C,0180,0800,0000,11C0  ;4251  NOP
U 11C0, 0818,0039,0D80,0800,0000,12E7  ;4252 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 11C1, 0818,0038,1980,0800,0000,1098  ;4253  D_K[ZERO]  ; Zero D to select Non-Interleaved Lower
U 1098, 0000,003D,0180,0800,0000,1263  ;4254 =00 CALL[SELECT.CNTRLR] ; Select Non-Interleaved lower
```

```
            ;4255  .PAGE
            ;4256 ;
            ;4257 ;////////////////////////////////////////////////////////////////////////////////
            ;4258 ;
            ;4259 ; Got a misconfiguration error when selecting Non-Interleave lower.  Call an
            ;4260 ; error.
            ;4261 ;
U 1099. 0000.003D.0D80.0800.0084.703C  ;4262  ERROR1[.3]
            ;4263 ;
            ;4264 ;////////////////////////////////////////////////////////////////////////////////
            ;4265 ;
            ;4266 ; ERROR LOGOUT:
            ;4267 ;
            ;4268 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4269 ;  Memory Test Address
            ;4270 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4271 ;       0 = Upper or Lower Present
            ;4272 ;
            ;4273 ;////////////////////////////////////////////////////////////////////////////////
            ;4274 ;
            ;4275 T.29.S1.NO.MISCONFIG:
U 109A. 0010.0038.0180.0968.0200.11C2  ;4276  VA_RC[0D]  ; Point to location 0
            ;4277 =
U 11C2. 0000.003D.8980.0800.0084.7293  ;4278 =0 SET.TRAP.MASK[.D] ; Enable Time out micro traps
U 11C3. 0000.003C.0180.C800.0000.139F  ;4279  D[LONG]_CACHE.P  ; Read location 0
U 139F. 0001.003C.0180.0980.0000.10A0  ;4280  RC[00]_D  ; Save read data in RC[00]
U 10A0. 0000.003D.0180.0800.0000.128D  ;4281 =00 CHECK.UTRAP  ; Was there a Timeout micro trap ?
U 10A1. 0000.003C.0180.0800.0000.10A4  ;4282  J/T.29.S1.NOTMO.1 ; No, continue testing
```

```
            ;4283  .PAGE
            ;4284 ;
            ;4285 ;////////////////////////////////////////////////////////////////////////////////
            ;4286 ;
            ;4287 ; Got a time out micro trap when reading the data from the test location.
            ;4288 ; Call an error.
            ;4289 ;
U 10A2. 0000.003D.0180.0800.0084.7044  ;4290  ERROR2[.8]
U 10A3. 0000.003C.0180.0800.0000.10A4  ;4291  NOP
            ;4292 ;
            ;4293 ;////////////////////////////////////////////////////////////////////////////////
            ;4294 ;
            ;4295 ; ERROR LOGOUT:
            ;4296 ;
            ;4297 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4298 ;  Memory Test Address
            ;4299 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4300 ;       0 = Upper or Lower Present
            ;4301 ;  Unused
            ;4302 ;  Unused
            ;4303 ;  Unused
            ;4304 ;  Expected Micro Trap Flags
            ;4305 ;  Received Micro Trap Flags
            ;4306 ;
            ;4307 ;////////////////////////////////////////////////////////////////////////////////
            ;4308 ;
            ;4309 =00
            ;4310 T.29.S1.NOTMO.1:
U 10A4. 0000.003D.0180.0800.0000.1280  ;4311  CHECK.INTERRUPT  ; Were there any unexpected interupts ?
U 10A5. 0000.003C.0180.0800.0000.10A8  ;4312  J/T.29.S1.NOINT.1 ; No, continue testing
```

```
            ;4313  .PAGE
            ;4314 ;
            ;4315 ;////////////////////////////////////////////////////////////////////////////////
            ;4316 ;
            ;4317 ; Got an unexpected interrupt when reading the data from the test location.
            ;4318 ; Call an error.
            ;4319 ;
U 10A6. 0000.003D.F580.0800.0084.703C  ;4320  ERROR1[.A]
U 10A7. 0000.003C.0180.0800.0000.10A8  ;4321  NOP
            ;4322 ;
            ;4323 ;////////////////////////////////////////////////////////////////////////////////
            ;4324 ;
            ;4325 ; ERROR LOGOUT:
            ;4326 ;
            ;4327 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4328 ;  Memory Test Address
            ;4329 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4330 ;       0 = Upper or Lower Present
            ;4331 ;  Unused
            ;4332 ;  Unused
            ;4333 ;  Unused
            ;4334 ;  ID[vector] Register
            ;4335 ;  ID[sbi.err] Register
            ;4336 ;  ID[fault] Register
            ;4337 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4338 ;       0 = No Interrupt Occurred
            ;4339 ;
            ;4340 ;////////////////////////////////////////////////////////////////////////////////
            ;4341 ;
            ;4342 =00
            ;4343 T.29.S1.NOINT.1:
U 10A8. 0000.003D.0180.0800.0000.1317  ;4344  CALL[DECODE]  ; Decode the data returned from test location
U 10A9. 0000.003C.0180.0800.0000.10AA  ;4345  NOP   ; Return1 if maximum errors
U 10AA. 0000.003C.0180.0800.0000.11C4  ;4346  J/T.29.CALL.ERDAT.2 ; Return2 if error but not maximum
U 10AB. 0000.003C.0180.0800.0000.11C7  ;4347  J/T.29.S1.CONTINUE ; Data returned was decoded as good, continue
```

```
            ;4348  .PAGE
            ;4349 ;
            ;4350 ;////////////////////////////////////////////////////////////////////////////////
            ;4351 ;
            ;4352 ; Data returned from test location was bad.  Call an error.
            ;4353 ;
            ;4354 =0
            ;4355 T.29.CALL.ERDAT.2:
U 11C4, 0000.003D.0180.0800.0000.12DE  ;4356  CALL[ERDAT]  ; Data is Bad, get the error logout data
U 11C5, 0000.003C.0180.0800.0000.11C6  ;4357  NOP
U 11C6, 0000.003D.6180.0800.0084.7044  ;4358 =0 ERROR2[.F]
            ;4359 ;
            ;4360 ;////////////////////////////////////////////////////////////////////////////////
            ;4361 ;
            ;4362 ; ERROR LOGOUT:
            ;4363 ;
            ;4364 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4365 ;  Memory Test Address
            ;4366 ;  Received Memory Data
            ;4367 ;  Expected Memory Data
            ;4368 ;  Received Memory Data
            ;4369 ;  Expected Memory Data
            ;4370 ;  Received Memory Data
            ;4371 ;  Expected Memory Data
            ;4372 ;  Received Memory Data
            ;4373 ;  Expected Memory Data
            ;4374 ;  Received Memory Data
            ;4375 ;  Expected Memory Data
            ;4376 ;  Received Memory Data
            ;4377 ;  Expected Memory Data
            ;4378 ;  FFFFFFFF
            ;4379 ;
            ;4380 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4381 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4382 ;        be output on the error then the last longword wrote will be the
            ;4383 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4384 ;
            ;4385 ;////////////////////////////////////////////////////////////////////////////////
            ;4386 ;
            ;4387 T.29.S1.CONTINUE:
U 11C7, 0010.0038.0180.0960.0010.13A0  ;4388  ALU_RC[0C],CLK.UBCC ; Move Controller(s) present flag to ALU
U 13A0, 0000.013C.0180.0800.0000.11C8  ;4389  Z?    ; Is the flag set or clear ?
U 11C8, 0000.003C.0180.0800.0000.11D2  ;4390 =0 J/T.29.S2  ; Flag is Set (both controllers present)
U 11C9, 0000.003C.0180.0800.0000.11CA  ;4391  NOP    ; Flag is clear (one controller present)
U 11CA, 0000.003D.0180.0800.0000.1259  ;4392 =0 ERLOOP   ; Loop on error from here
U 11CB, 0000.003C.0180.0800.0000.11CC  ;4393  NOP
U 11CC, 0000.003D.0180.0800.0000.107C  ;4394 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11CD, 0818.0038.1180.0800.0000.10AC  ;4395  D_K[.4]    ; Set-up to select Internal Interleave
U 10AC, 0000.003D.0180.0800.0000.1263  ;4396 =00 CALL[SELECT.CNTRLR] ; Select internal Interleave
U 10AD, 0000.003C.0180.0800.0000.10AF  ;4397  J/T.29.S1.CHK.BIT16 ; OK. Got Misconfig, go check bit 16 in CNFG A
```

```
            ;4398  .PAGE
            ;4399 ;
            ;4400 ;////////////////////////////////////////////////////////////////////////////////
            ;4401 ;
            ;4402 ; Didn't get an misconfiguration error with the Upper Controller missing in
            ;4403 ; Internal Interleave mode.  Should have got this error.  Call an error.
            ;4404 ;
            ;4405 =0
            ;4406 T.29.S1.NO.MISCNFG:
U 10AE, 0000,003D,0D80,0800,0084,7044  ;4407  ERROR2[.3]
            ;4408 ;
            ;4409 ;////////////////////////////////////////////////////////////////////////////////
            ;4410 ;
            ;4411 ; ERROR LOGOUT:
            ;4412 ;
            ;4413 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4414 ;  Memory Test Address
            ;4415 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4416 ;       0 = Upper or Lower Present
            ;4417 ;
            ;4418 ;////////////////////////////////////////////////////////////////////////////////
            ;4419 ;
            ;4420 T.29.S1.CHK.BIT16:
U 10AF, 0010,0038,0180,0970,0200,13A1  ;4421  VA_RC[0E]  ; Load VA with address of CNFG A
U 13A1, 0000,003C,0180,C800,0000,13A2  ;4422  D[LONG]_CACHE.P  ; Read CNFG A
U 13A2, 0000,003C,01E0,0800,0000,13A3  ;4423  Q_D  ; Move CNFG A data to the Q register
U 13A3, 0001,203C,0180,09D8,0000,13A4  ;4424  RC[0B]_Q  ; Save in RC[0B] in case error
U 13A4, 0000,003C,6580,0800,0084,73A5  ;4425  SC_K[.10]  ; Set-up SC to mask bit 16
U 13A5, 0803,001C,0180,0800,0000,13A6  ;4426  D_NOT.MASK  ; Assert bit 16 in the D register
            ;4427  ALU_D.AND.Q,  ; AND masked bit 16 with CNFG A data
U 13A6, 001D,0034,0180,0800,0010,13A7  ;4428  CLK.UBCC ; ...
U 13A7, 0000,013C,0180,0800,0000,11CE  ;4429  Z?  ; Is bit 16 set or clear ?
U 11CE, 0000,003C,0180,0800,0000,1138  ;4430 =0 J/T.29.CHECK.NEXT ; Bit 16 was set, go to subest 2
U 11CF, 0000,003C,0180,0800,0000,11D0  ;4431  NOP  ; Bit 16 was clear, Call an error
```

```
            ;4432  .PAGE
            ;4433 ;
            ;4434 ;////////////////////////////////////////////////////////////////////////////////
            ;4435 ;
            ;4436 ; Bit 16 ( Misconfiguration Error in Upper Controller Memory ) in CNFG A was
            ;4437 ; not set and it should have been.  Call an error.
            ;4438 ;
U 11D0, 0000,003D,1180,0800,0084,7044  ;4439 =0 ERROR2[.4]
U 11D1, 0000,003C,0180,0800,0000,1138  ;4440  J/T.29.CHECK.NEXT
            ;4441 ;
            ;4442 ;////////////////////////////////////////////////////////////////////////////////
            ;4443 ;
            ;4444 ; ERROR LOGOUT:
            ;4445 ;
            ;4446 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4447 ;  Memory Test Address
            ;4448 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4449 ;       0 = Upper or Lower Present
            ;4450 ;  Configuration Register A
            ;4451 ;
            ;4452 ;////////////////////////////////////////////////////////////////////////////////
            ;4453 ;
            ;4454 ;
            ;4455 ;================================================================================
            ;4456 ;
            ;4457 ; Subtest 2
            ;4458 ;
            ;4459 =0
            ;4460 T.29.S2:
U 11D2, 0000,003D,0180,0800,0000,1261  ;4461  SUBTEST   ; Increment Subtest counter
U 11D3, 0000,003C,1980,0800,1404,73A8  ;4462  STATE_K[ZERO]  ; Clear the State register used by decode/erdat
U 13A8, 0003,003C,0180,09E8,0000,13A9  ;4463  RC[0D]_0  ; Load RC[0D] with test address
U 13A9, 0018,0038,0580,09A0,0000,11D4  ;4464  RC[04]_K[.1]  ; Set pattern count to 1
U 11D4, 0000,003D,0180,0800,0000,12F3  ;4465 =0 CALL[ENCODE]  ; Encode the pattern
U 11D5, 0000,003C,0180,0800,0000,11D6  ;4466  NOP
U 11D6, 0000,003D,0180,0800,0000,1259  ;4467 =0 ERLOOP   ; Loop on error from here
U 11D7, 0000,003C,0180,0800,0000,11D8  ;4468  NOP
U 11D8, 0000,003D,0180,0800,0000,107C  ;4469 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11D9, 0000,003C,0180,0800,0000,11DA  ;4470  NOP
U 11DA, 0818,0039,0D80,0800,0000,12E7  ;4471 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 11DB, 0000,003C,0180,0800,0000,13AA  ;4472  NOP
U 13AA, 0818,0038,0980,0800,0000,10B0  ;4473  D_K[.2]   ; Move value to specify Non-Interleave Upper
            ;4474      ; to the D register for the next call
U 10B0, 0000,003D,0180,0800,0000,1263  ;4475 =00 CALL[SELECT.CNTRLR] ; Select Non-Interleave Upper
```

```
            ;4476  .PAGE
            ;4477 ;
            ;4478 ;////////////////////////////////////////////////////////////////////////////////
            ;4479 ;
            ;4480 ; Got a misconfiguration error when selecting Non-Interleave mode.  Call an
            ;4481 ; error
            ;4482 ;
            ;4483 T.29.S2.CONFIG.ERR:
U 10B1, 0000,003D.0D80,0800,0084,703C  ;4484  ERROR1[.3]
            ;4485 ;
            ;4486 ;////////////////////////////////////////////////////////////////////////////////
            ;4487 ;
            ;4488 ; ERROR LOGOUT:
            ;4489 ;
            ;4490 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4491 ;  Memory Test Address
            ;4492 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4493 ;       0 = Upper or Lower Present
            ;4494 ;
            ;4495 ;////////////////////////////////////////////////////////////////////////////////
            ;4496 ;
            ;4497 T.29.S2.NO.CONFIG.ERR:
U 10B2, 0010,0038,0180,0968,0200,10B3  ;4498  VA_RC[0D]  ; Load VA with test address
U 10B3, 0810,0038,0180,0908,0000,13AB  ;4499  D_RC[01]  ; Load D with encoded vector
U 13AB, 0000,003C,0180,A800,0000,10B4  ;4500  CACHE.P_D[LONG]  ; Write the coded vector to the location
U 10B4, 0000,003D,0180,0800,0000,1280  ;4501 =00 CHECK.INTERRUPT  ; Were there any unexpected interrupts ?
U 10B5, 0000,003C,0180,0800,0000,10B8  ;4502  J/T.29.S2.CHK.TR ; No, continue with test
```

```
            ;4503  .PAGE
            ;4504 ;
            ;4505 ;////////////////////////////////////////////////////////////////////////////////
            ;4506 ;
            ;4507 ; Got an unexpected interrupt when writing the encoded value out to memory.
            ;4508 ; Call an error
            ;4509 ;
U 10B6, 0000,003D,F580,0800,0084,703C  ;4510  ERROR1[.A]
U 10B7, 0000,003C,0180,0800,0000,113B  ;4511  J/T.29.EXIT
            ;4512 ;
            ;4513 ;////////////////////////////////////////////////////////////////////////////////
            ;4514 ;
            ;4515 ; ERROR LOGOUT:
            ;4516 ;
            ;4517 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4518 ;  Memory Test Address
            ;4519 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4520 ;       0 = Upper or Lower Present
            ;4521 ;  Unused
            ;4522 ;  Unused
            ;4523 ;  Unused
            ;4524 ;  ID[vector] Register
            ;4525 ;  ID[sbi.err] Register
            ;4526 ;  ID[fault] Register
            ;4527 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4528 ;       0 = No Interrupt Occurred
            ;4529 ;
            ;4530 ;////////////////////////////////////////////////////////////////////////////////
            ;4531 ;
            ;4532 =00
            ;4533 T.29.S2.CHK.TR:
U 10B8, 0000,003D,0180,0800,0000,12ED  ;4534  CALL[ODD.EVEN.TR] ; Is this memory at odd or even TR ?
            ;4535  RC[0D]_K[.8],  ; Even TR
U 10B9, 0018,0038,0180,09E8,0000,11DC  ;4536  J/T.29.S2.SEC.2
U 10BA, 0018,0038,1980,09E8,0000,11DC  ;4537  RC[0D]_K[ZERO]  ; Odd TR
            ;4538 =
            ;4539 =0
            ;4540 T.29.S2.SEC.2:
U 11DC, 0000,003D,0180,0800,0000,1259  ;4541  ERLOOP   ; Loop on error from here
U 11DD, 0000,003C,0180,0800,0000,11DE  ;4542  NOP
U 11DE, 0000,003D,0180,0800,0000,107C  ;4543 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11DF, 0000,003C,0180,0800,0000,11E0  ;4544  NOP
U 11E0, 0818,0039,0D80,0800,0000,12E7  ;4545 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 11E1, 0818,0038,0D80,0800,0000,10BC  ;4546  D_K[.3]   ; Load D with 3 for Externally Interleaved
U 10BC, 0000,003D,0180,0800,0000,1263  ;4547 =00 CALL[SELECT.CNTRLR] ; Select Externally Interleaved Upper
```

```
            ;4548  .PAGE
            ;4549 ;
            ;4550 ;////////////////////////////////////////////////////////////////////////////////
            ;4551 ;
            ;4552 ; Got a misconfiguration error when selecting Non-Interleave mode.  Call an
            ;4553 ; error
            ;4554 ;
            ;4555 T.29.S2.CONFIG.ERR.2:
U 10BD, 0000,003D,0D80,0800,0084,703C  ;4556  ERROR1[.3]
U 10BE, 0000,003C,0180,0800,0000,11E2  ;4557  NOP
            ;4558 =
            ;4559 ;
            ;4560 ;////////////////////////////////////////////////////////////////////////////////
            ;4561 ;
            ;4562 ; ERROR LOGOUT:
            ;4563 ;
            ;4564 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4565 ;  Memory Test Address  0 = Odd TR 8 = Even TR
            ;4566 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4567 ;       0 = Upper or Lower Present
            ;4568 ;
            ;4569 ;////////////////////////////////////////////////////////////////////////////////
            ;4570 ;
            ;4571 =0
            ;4572 T.29.S2.CONT.SEC.2:
U 11E2, 0000,003D,8980,0800,0084,7293  ;4573  SET.TRAP.MASK[.D] ; Enable Time out micro traps
U 11E3, 0010,0038,0180,0968,0200,13AC  ;4574  VA_RC[0D]  ; Point VA to the test location
U 13AC, 0000,003C,0180,C800,0000,13AD  ;4575  D[LONG]_CACHE.P  ; Read the test location
U 13AD, 0001,003C,0180,0980,0000,10C0  ;4576  RC[00]_D  ; Save the read data in RC[00]
U 10C0, 0000,003D,0180,0800,0000,128D  ;4577 =00 CHECK.UTRAP  ; Check for any micro traps
U 10C1, 0000,003C,0180,0800,0000,10C4  ;4578  J/T.29.S2.CHK.INT.1 ; No, go check for any unexpected interrupts
```

```
            ;4579  .PAGE
            ;4580 ;
            ;4581 ;////////////////////////////////////////////////////////////////////////////
            ;4582 ;
            ;4583 ; Got a time out micro trap when reading the data from the test location.
            ;4584 ; Call an error.
            ;4585 ;
U 10C2, 0000,003D,0180,0800,0084,7044  ;4586  ERROR2[.8]
U 10C3, 0000,003C,0180,0800,0000,10C4  ;4587  NOP
            ;4588 ;
            ;4589 ;////////////////////////////////////////////////////////////////////////////
            ;4590 ;
            ;4591 ; ERROR LOGOUT:
            ;4592 ;
            ;4593 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4594 ;  Memory Test Address
            ;4595 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4596 ;       0 = Upper or Lower Present
            ;4597 ;  Unused
            ;4598 ;  Unused
            ;4599 ;  Unused
            ;4600 ;  Expected Micro Trap Flags
            ;4601 ;  Received Micro Trap Flags
            ;4602 ;
            ;4603 ;////////////////////////////////////////////////////////////////////////////
            ;4604 ;
            ;4605 =00
            ;4606 T.29.S2.CHK.INT.1:
U 10C4, 0000,003D,0180,0800,0000,1280  ;4607  CHECK.INTERRUPT  ; Any unexpected Interrupts ?
U 10C5, 0000,003C,0180,0800,0000,10C8  ;4608  J/T.29.S2.DO.DECODE ; No, no go decode the data
```

```
            ;4609  .PAGE
            ;4610 ;
            ;4611 ;////////////////////////////////////////////////////////////////////////////////
            ;4612 ;
            ;4613 ; Got an unexpected interrupt when reading the data from the test location.
            ;4614 ; Call an error.
            ;4615 ;
U 10C6, 0000,003D,F580,0800,0084,703C  ;4616  ERROR1[.A]
U 10C7, 0000,003C,0180,0800,0000,10C8  ;4617  NOP
            ;4618 ;
            ;4619 ;////////////////////////////////////////////////////////////////////////////////
            ;4620 ;
            ;4621 ; ERROR LOGOUT:
            ;4622 ;
            ;4623 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4624 ;  Memory Test Address
            ;4625 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4626 ;       0 = Upper or Lower Present
            ;4627 ;  Unused
            ;4628 ;  Unused
            ;4629 ;  Unused
            ;4630 ;  ID[vector] Register
            ;4631 ;  ID[sbi.err] Register
            ;4632 ;  ID[fault] Register
            ;4633 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4634 ;       0 = No Interrupt Occurred
            ;4635 ;
            ;4636 ;////////////////////////////////////////////////////////////////////////////////
            ;4637 ;
            ;4638 =00
            ;4639 T.29.S2.DO.DECODE:
U 10C8, 0000,003D,0180,0800,0000,1317  ;4640  CALL[DECODE]  ; Check the data returned
U 10C9, 0000,003C,0180,0800,0000,10CA  ;4641  NOP    ; Return1 if maximum errors
U 10CA, 0000,003C,0180,0800,0000,11E4  ;4642  J/T.29.CALL.ERDAT.3 ; Return2 if error but not maximum
U 10CB, 0000,003C,0180,0800,0000,11E8  ;4643  J/T.29.S2.READ.NXM ; Data returned was good, try reading a
            ;4644     ; Non Existent Memory location
```

```
            ;4645  .PAGE
            ;4646 ;
            ;4647 ;////////////////////////////////////////////////////////////////////////////////
            ;4648 ;
            ;4649 ; Data that was decoded from the read of the test location was bad.  Call an
            ;4650 ; error.
            ;4651 ;
            ;4652 =0
            ;4653 T.29.CALL.ERDAT.3:
U 11E4, 0000.003D.0180.0800.0000.12DE  ;4654  CALL[ERDAT]  ; Get the logout data
U 11E5, 0000.003C.0180.0800.0000.11E6  ;4655  NOP
U 11E6, 0000.003D.6180.0800.0084.703C  ;4656 =0 ERROR1[.F]
U 11E7, 0000.003C.0180.0800.0000.11E8  ;4657  NOP
            ;4658 ;
            ;4659 ;////////////////////////////////////////////////////////////////////////////////
            ;4660 ;
            ;4661 ; ERROR LOGOUT:
            ;4662 ;
            ;4663 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4664 ;  Memory Test Address
            ;4665 ;  Received Memory Data
            ;4666 ;  Expected Memory Data
            ;4667 ;  Received Memory Data
            ;4668 ;  Expected Memory Data
            ;4669 ;  Received Memory Data
            ;4670 ;  Expected Memory Data
            ;4671 ;  Received Memory Data
            ;4672 ;  Expected Memory Data
            ;4673 ;  Received Memory Data
            ;4674 ;  Expected Memory Data
            ;4675 ;  Received Memory Data
            ;4676 ;  Expected Memory Data
            ;4677 ;  FFFFFFFF
            ;4678 ;
            ;4679 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4680 ;       terminated by a xFFFFFFFF.  If there are more errors then can
            ;4681 ;       be output on the error then the last longword wrote will be the
            ;4682 ;       xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4683 ;
            ;4684 ;////////////////////////////////////////////////////////////////////////////////
            ;4685 ;
            ;4686 =0
            ;4687 T.29.S2.READ.NXM:
U 11E8, 0000.003D.0180.0800.0000.1259  ;4688  ERLOOP   ; Loop on error from here
U 11E9, 0010.0038.0180.0968.0200.11EA  ;4689  VA_ALU,ALU_RC[0D] ; Point to a non-existent memory location
            ;4690 =0 CALL[SBI.INIT],  ; Initialize the SBI
U 11EA, 0000.003D.0180.0803.0000.107C  ;4691   VA_VA+4  ; VA should be pointing to location
U 11EB, 0000.003C.0180.0803.0000.11EC  ;4692  VA_VA+4   ; ...8 or C
U 11EC, 0000.003D.8980.0800.0084.7293  ;4693 =0 SET.TRAP.MASK[.D] ; Enable time out micro traps
U 11ED, 0000.003C.0180.C800.0000.10CC  ;4694  D[LONG]_CACHE.P  ; Try reading the nonexistent memory location
U 10CC, 0000.003D.0180.0800.0000.128D  ;4695 =00 CHECK.UTRAP  ; Was there a time out micro trap
U 10CD, 0000.003C.0180.0800.0000.11EE  ;4696  J/T.29.S2.NOTMO.ERR ; No, there should have been, call an error
U 10CE, 0000.003C.0180.0800.0000.11EF  ;4697  J/T.29.S2.DO.ENCODE ; Yes, continue with test
            ;4698 =
```

```
            ;4699  .PAGE
            ;4700 ;
            ;4701 ;////////////////////////////////////////////////////////////////////////////////
            ;4702 ;
            ;4703 ; Didn't get a time out micro trap when reading a non-existent memory location.
            ;4704 ; Call an error.  Should have got time out.
            ;4705 ;
            ;4706 =0
            ;4707 T.29.S2.NOTMO.ERR:
U 11EE, 0000.003D.0180.0800.0084.7044  ;4708  ERROR2[.8]
            ;4709 ;
            ;4710 ;////////////////////////////////////////////////////////////////////////////////
            ;4711 ;
            ;4712 ; ERROR LOGOUT:
            ;4713 ;
            ;4714 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4715 ;  Memory Test Address
            ;4716 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4717 ;       0 = Upper or Lower Present
            ;4718 ;  Unused
            ;4719 ;  Unused
            ;4720 ;  Unused
            ;4721 ;  Expected Micro Trap Flags
            ;4722 ;  Received Micro Trap Flags
            ;4723 ;
            ;4724 ;////////////////////////////////////////////////////////////////////////////////
            ;4725 ;
            ;4726 T.29.S2.DO.ENCODE:
U 11EF, 0810.0038.0180.0920.0000.13AE  ;4727  D_RC[04]  ; Move pattern select (RC[04]) to the D reg
U 13AE, 0019.0014.0980.09A0.0000.11F0  ;4728  RC[04]_D+K[.2]  ; Add 2 to select new pattern
U 11F0, 0000.003D.0180.0800.0000.12F3  ;4729 =0 CALL[ENCODE]  ; Encode the new pattern
U 11F1, 0000.003C.0180.0800.0000.11F2  ;4730  NOP
U 11F2, 0000.003D.0180.0800.0000.1259  ;4731 =0 ERLOOP  ; Loop on error from here
U 11F3, 0000.003C.0180.0800.0000.11F4  ;4732  NOP
U 11F4, 0000.003D.0180.0800.0000.107C  ;4733 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11F5, 0000.003C.0180.0800.0000.11F6  ;4734  NOP
U 11F6, 0818.0039.0D80.0800.0000.12E7  ;4735 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 11F7, 0010.0038.0180.0968.0200.13AF  ;4736  VA_RC[0D]  ; Load VA with 0 for memory location 0
U 13AF, 0810.0038.0180.0908.0000.13B0  ;4737  D_RC[01]  ; Move pattern returned from Encode to D reg
U 13B0, 0000.003C.0180.A800.0000.10D0  ;4738  CACHE.P_D[LONG]  ; Write the pattern to memory
U 10D0, 0000.003D.0180.0800.0000.1280  ;4739 =00 CHECK.INTERRUPT  ; Any unexpected interrupts on the write ?
U 10D1, 0000.003C.0180.0800.0000.10D3  ;4740  J/T.29.S2.CHK.READ ; No, get ready to read the data back
```

```
            ;4741  .PAGE
            ;4742 ;
            ;4743 ;////////////////////////////////////////////////////////////////////////////////
            ;4744 ;
            ;4745 ; Got an unexpected interrupt when writing the data to the test location.
            ;4746 ; Call an error.
            ;4747 ;
U 10D2, 0000,003D,F580,0800,0084,703C  ;4748  ERROR1[.A]
            ;4749 ;
            ;4750 ;////////////////////////////////////////////////////////////////////////////////
            ;4751 ;
            ;4752 ; ERROR LOGOUT:
            ;4753 ;
            ;4754 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4755 ;  Memory Test Address
            ;4756 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4757 ;       0 = Upper or Lower Present
            ;4758 ;  Unused
            ;4759 ;  Unused
            ;4760 ;  Unused
            ;4761 ;  ID[vector] Register
            ;4762 ;  ID[sbi.err] Register
            ;4763 ;  ID[fault] Register
            ;4764 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4765 ;       0 = No Interrupt Occurred
            ;4766 ;
            ;4767 ;////////////////////////////////////////////////////////////////////////////////
            ;4768 ;
            ;4769 T.29.S2.CHK.READ:
U 10D3, 0003,003C,0180,09E8,0000,11F8  ;4770  RC[0D]_0  ; Set RC[0D] to 0, location to be accessed
U 11F8, 0000,003D,0180,0800,0000,1259  ;4771 =0 ERLOOP   ; Loop on error from here
U 11F9, 0000,003C,0180,0800,0000,11FA  ;4772  NOP
U 11FA, 0000,003D,0180,0800,0000,107C  ;4773 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11FB, 0900,003C,0180,0800,0000,11FC  ;4774  NOP
U 11FC, 0818,0039,0D80,0800,0000,12E7  ;4775 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 11FD, 0818,0038,0980,0800,0000,10D4  ;4776  D_K[.2]   ; Zero D to select Non-Interleaved Upper
U 10D4, 0000,003D,0180,0800,0000,1263  ;4777 =00 CALL[SELECT.CNTRLR] ; Select Non-Interleaved Upper
```

```
            ;4778  .PAGE
            ;4779 ;
            ;4780 ;////////////////////////////////////////////////////////////////////////////
            ;4781 ;
            ;4782 ; Got a misconfiguration error when selecting Non-Interleave lower.  Call an
            ;4783 ; error.
            ;4784 ;
U 10D5, 0000,003D,0D80,0800,0084,703C  ;4785  ERROR1[.3]
            ;4786 ;
            ;4787 ;////////////////////////////////////////////////////////////////////////////
            ;4788 ;
            ;4789 ; ERROR LOGOUT:
            ;4790 ;
            ;4791 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4792 ;  Memory Test Address
            ;4793 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4794 ;       0 = Upper or Lower Present
            ;4795 ;
            ;4796 ;////////////////////////////////////////////////////////////////////////////
            ;4797 ;
            ;4798 T.29.S2.NO.MISCONFIG:
U 10D6, 0010,0038,0180,0968,0200,11FE  ;4799  VA_RC[0D]  ; Point to location 0
            ;4800 =
U 11FE, 0000,003D,8980,0800,0084,7293  ;4801 =0 SET.TRAP.MASK[.D] ; Enable Time out micro traps
U 11FF, 0000,003C,0180,C800,0000,13B1  ;4802  D[LONG]_CACHE.P  ; Read location 0
U 13B1, 0001,003C,0180,0980,0000,10D8  ;4803  RC[00]_D  ; Save read data in RC[00]
U 10D8, 0000,003D,0180,0800,0000,128D  ;4804 =00 CHECK.UTRAP  ; Was there a Timeout micro trap ?
U 10D9, 0000,003C,0180,0800,0000,10DC  ;4805  J/T.29.S2.NOTMO.1 ; No, continue testing
```

```
            ;4806  .PAGE
            ;4807  ;
            ;4808  ;////////////////////////////////////////////////////////////////////////////////
            ;4809  ;
            ;4810  ; Got a time out micro trap when reading the data from the test location.
            ;4811  ; Call an error.
            ;4812  ;
U 10DA, 0000,003D,0180,0800,0084,7044  ;4813  ERROR2[.8]
U 10DB, 0000,003C,0180,0800,0000,10DC  ;4814  NOP
            ;4815  ;
            ;4816  ;////////////////////////////////////////////////////////////////////////////////
            ;4817  ;
            ;4818  ; ERROR LOGOUT:
            ;4819  ;
            ;4820  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4821  ;  Memory Test Address
            ;4822  ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4823  ;       0 = Upper or Lower Present
            ;4824  ;  Unused
            ;4825  ;  Unused
            ;4826  ;  Unused
            ;4827  ;  Expected Micro Trap Flags
            ;4828  ;  Received Micro Trap Flags
            ;4829  ;
            ;4830  ;////////////////////////////////////////////////////////////////////////////////
            ;4831  ;
            ;4832  =00
            ;4833  T.29.S2.NOTMO.1:
U 10DC, 0000,003D,0180,0800,00D0,1280  ;4834  CHECK.INTERRUPT  ; Were there any unexpected interupts ?
U 10DD, 0000,003C,0180,0800,0000,10E0  ;4835  J/T.29.S2.NOINT.1 ; No, continue testing
```

```
            ;4836  .PAGE
            ;4837  ;
            ;4838  ;////////////////////////////////////////////////////////////////////////////////
            ;4839  ;
            ;4840  ; Got an unexpected interrupt when reading the data from the test location.
            ;4841  ; Call an error.
            ;4842  ;
U 10DE, 0000,003D,F580,0800,0084,703C  ;4843  ERROR1[.A]
U 10DF, 0000,003C,0180,0800,0000,10E0  ;4844  NOP
            ;4845  ;
            ;4846  ;////////////////////////////////////////////////////////////////////////////////
            ;4847  ;
            ;4848  ; ERROR LOGOUT:
            ;4849  ;
            ;4850  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4851  ;  Memory Test Address
            ;4852  ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4853  ;       0 = Upper or Lower Present
            ;4854  ;  Unused
            ;4855  ;  Unused
            ;4856  ;  Unused
            ;4857  ;  ID[vector] Register
            ;4858  ;  ID[sbi.err] Register
            ;4859  ;  ID[fault] Register
            ;4860  ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;4861  ;       0 = No Interrupt Occurred
            ;4862  ;
            ;4863  ;////////////////////////////////////////////////////////////////////////////////
            ;4864  ;
            ;4865  =00
            ;4866  T.29.S2.NOINT.1:
U 10E0, 0000,003D,0180,0800,0000,1317  ;4867  CALL[DECODE]  ; Decode the data returned from test location
U 10E1, 0000,003C,0180,0800,0000,10E2  ;4868  NOP    ; Return1 if maximum errors
U 10E2, 0000,003C,0180,0800,0000,1200  ;4869  J/T.29.CALL.ERDAT.4 ; Return2 if error but not maximum
U 10E3, 0000,003C,0180,0800,0000,1203  ;4870  J/T.29.S2.CONTINUE ; Data returned was decoded as good, continue
```

```
            ;4871   .PAGE
            ;4872 ;
            ;4873 ;////////////////////////////////////////////////////////////////////////////////
            ;4874 ;
            ;4875 ; Data returned from test location was bad.  Call an error.
            ;4876 ;
            ;4877 =0
            ;4878 T.29.CALL.ERDAT.4:
U 1200, 0000,003D,0180,0800,0000,12DE   ;4879  CALL[ERDAT]  ; Data is Bad, get the error logout data
U 1201, 0000,003C,0180,0800,0000,1202   ;4880  NOP
U 1202, 0000,003D,6180,0800,0084,703C   ;4881 =0 ERROR1[.F]
            ;4882 ;
            ;4883 ;////////////////////////////////////////////////////////////////////////////////
            ;4884 ;
            ;4885 ; ERROR LOGOUT:
            ;4886 ;
            ;4887 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4888 ;  Memory Test Address
            ;4889 ;  Received Memory Data
            ;4890 ;  Expected Memory Data
            ;4891 ;  Received Memory Data
            ;4892 ;  Expected Memory Data
            ;4893 ;  Received Memory Data
            ;4894 ;  Expected Memory Data
            ;4895 ;  Received Memory Data
            ;4896 ;  Expected Memory Data
            ;4897 ;  Received Memory Data
            ;4898 ;  Expected Memory Data
            ;4899 ;  Received Memory Data
            ;4900 ;  Expected Memory Data
            ;4901 ;  FFFFFFFF
            ;4902 ;
            ;4903 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4904 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4905 ;        be output on the error then the last longword wrote will be the
            ;4906 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4907 ;
            ;4908 ;////////////////////////////////////////////////////////////////////////////////
            ;4909 ;
            ;4910 T.29.S2.CONTINUE:
U 1203, 0010,0038,0180,0960,0010,13B2   ;4911  ALU_RC[0C],CLK.UBCC ; Move Controller(s) present flag to ALU
U 13B2, 0000,013C,0180,0800,0000,1204   ;4912  Z?    ; Is the flag set or clear ?
U 1204, 0000,003C,0180,0800,0000,120E   ;4913 =0 J/T.29.S3  ; Flag is Set (both controllers present)
U 1205, 0000,003C,0180,0800,0000,1206   ;4914  NOP    ; Flag is clear (one controller present)
U 1206, 0000,003D,0180,0800,0000,1259   ;4915 =0 ERLOOP    ; Loop on error from here
U 1207, 0000,003C,0180,0800,0000,1208   ;4916  NOP
U 1208, 0000,003D,0180,0800,0000,107C   ;4917 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1209, 0818,0038,1180,0800,0000,10E4   ;4918  D_K[.4]    ; Set-up to select Internal Interleave
U 10E4, 0000,003D,0180,0800,0000,1263   ;4919 =00 CALL[SELECT.CNTRLR] ; Select internal Interleave
U 10E5, 0000,003C,0180,0800,0000,10E7   ;4920  J/T.29.S2.CHK.BIT15 ; OK. Got Misconfig, go check bit 15 in CNFG A
```

```
            ;4921  .PAGE
            ;4922 ;
            ;4923 ;////////////////////////////////////////////////////////////////////////////
            ;4924 ;
            ;4925 ; Didn't get an misconfiguration error with the Upper Controller missing in
            ;4926 ; Internal Interleave mode.  Should have got this error.  Call an error.
            ;4927 ;
            ;4928 =0
            ;4929 T.29.S2.NO.MISCNFG:
U 10E6, 0000,003D,0D80,0800,0084,7044  ;4930  ERROR2[.3]
            ;4931 ;
            ;4932 ;////////////////////////////////////////////////////////////////////////////
            ;4933 ;
            ;4934 ; ERROR LOGOUT:
            ;4935 ;
            ;4936 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4937 ;  Memory Test Address
            ;4938 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4939 ;       0 = Upper or Lower Present
            ;4940 ;
            ;4941 ;////////////////////////////////////////////////////////////////////////////
            ;4942 ;
            ;4943 T.29.S2.CHK.BIT15:
U 10E7, 0010,0038,0180,0970,0200,13B3  ;4944  VA_RC[0E]  ; Load VA with address of CNFG A
U 13B3, 0000,003C,0180,C800,0000,13B4  ;4945  D[LONG]_CACHE.P  ; Read CNFG A
U 13B4, 0000,003C,01E0,0800,0000,13B5  ;4946  Q_D  ; Move CNFG A data to the Q register
U 13B5, 0001,203C,0180,09D8,0000,13B6  ;4947  RC[0B]_Q  ; Save CNFG A in RC[0B] in case error
U 13B6, 0000,003C,6180,0800,0084,73B7  ;4948  SC_K[.F]  ; Set-up SC to mask bit 15
U 13B7, 0803,001C,0180,0800,0000,13B8  ;4949  D_NOT.MASK  ; Assert bit 15 in the D register
            ;4950  ALU_D.AND.Q,  ; AND masked bit 15 with CNFG A data
U 13B8, 001D,0034,0180,0800,0010,13B9  ;4951  CLK.UBCC ; ...
U 13B9, 0000,013C,0180,0800,0000,120A  ;4952  Z?  ; Is bit 15 set or clear ?
U 120A, 0000,003C,0180,0800,0000,1138  ;4953 =0 J/T.29.CHECK.NEXT ; Bit 15 was set, go to subest 2
U 120B, 0000,003C,0180,0800,0000,120C  ;4954  NOP  ; Bit 15 was clear, Call an error
```

```
            ;4955  .PAGE
            ;4956 ;
            ;4957 ;////////////////////////////////////////////////////////////////////////////////
            ;4958 ;
            ;4959 ; Bit 15 ( Misconfiguration Error in Lower Controller Memory ) in CNFG A was
            ;4960 ; not set and it should have been.  Call an error.
            ;4961 ;
U 120C, 0000,003D,1180,0800,0084,7044  ;4962 =0 ERROR2[.4]
U 120D, 0000,003C,0180,0800,0000,1138  ;4963  J/T.29.CHECK.NEXT
            ;4964 ;
            ;4965 ;////////////////////////////////////////////////////////////////////////////////
            ;4966 ;
            ;4967 ; ERROR LOGOUT:
            ;4968 ;
            ;4969 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4970 ;  Memory Test Address
            ;4971 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;4972 ;       0 = Upper or Lower Present
            ;4973 ;  Configuration Register A
            ;4974 ;
            ;4975 ;////////////////////////////////////////////////////////////////////////////////
            ;4976 ;
            ;4977 ;
            ;4978 ;================================================================================
            ;4979 ;
            ;4980 ; Subtest 3
            ;4981 ;
            ;4982 =0
            ;4983 T.29.S3:
U 120E, 0000,003D,0180,0800,0000,1261  ;4984  SUBTEST   ; Increment Subtest counter
U 120F, 0000,003C,1980,0800,1404,73BA  ;4985  STATE_K[ZERO]  ; Clear the State register used by decode/erdat
U 13BA, 0003,003C,0180,09E8,0000,13BB  ;4986  RC[0D]_0  ; Load RC[0D] with test address
U 13BB, 0018,0038,0580,09A0,0000,1210  ;4987  RC[04]_K[.1]  ; Set pattern count to 1
U 1210, 0000,003D,0180,0800,0000,12F3  ;4988 =0 CALL[ENCODE]  ; Encode the pattern
U 1211, 0000,003C,0180,0800,0000,1212  ;4989  NOP
U 1212, 0000,003D,0180,0800,0000,1259  ;4990 =0 ERLOOP   ; Loop on error from here
U 1213, 0000,003C,0180,0800,0000,1214  ;4991  NOP
U 1214, 0000,003D,0180,0800,0000,107C  ;4992 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1215, 0000,003C,0180,0800,0000,1216  ;4993  NOP
U 1216, 0818,0039,0D80,0800,0000,12E7  ;4994 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 1217, 0818,0038,1180,0800,0000,10E8  ;4995  D_K[.4]   ; Move value to specify Internal Interleave
            ;4996      ; to the D register for the next call
U 10E8, 0000,003D,0180,0800,0000,1263  ;4997 =00 CALL[SELECT.CNTRLR] ; Select Internal Interleave mode
```

```
            ;4998  .PAGE
            ;4999 ;
            ;5000 ;////////////////////////////////////////////////////////////////////////////
            ;5001 ;
            ;5002 ; Got a misconfiguration error when selecting Non-Interleave mode.  Call an
            ;5003 ; error
            ;5004 ;
            ;5005 T.29.S3.CONFIG.ERR:
U 10E9, 0000,003D,0D80,0800,0084,703C  ;5006  ERROR1[.3]
            ;5007 ;
            ;5008 ;////////////////////////////////////////////////////////////////////////////
            ;5009 ;
            ;5010 ; ERROR LOGOUT:
            ;5011 ;
            ;5012 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5013 ;  Memory Test Address
            ;5014 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5015 ;       0 = Upper or Lower Present
            ;5016 ;
            ;5017 ;////////////////////////////////////////////////////////////////////////////
            ;5018 ;
            ;5019 T.29.S3.NO.CONFIG.ERR:
U 10EA, 0010,0038,0180,0968,0200,10EB  ;5020  VA_RC[0D]  ; Load VA with test address
U 10EB, 0810,0038,0180,0908,0000,13BC  ;5021  D_RC[01]  ; Load D with encoded vector
U 13BC, 0000,003C,0180,A800,0000,10EC  ;5022  CACHE.P_D[LONG]  ; Write the coded vector to the location
U 10EC, 0000,003D,0180,0800,0000,1280  ;5023 =00 CHECK.INTERRUPT  ; Were there any unexpected interrupts ?
U 10ED, 0000,003C,0180,0800,0000,10EF  ;5024  J/T.29.S3.DO.SEC ; No, continue with test
```

```
            ;5025  .PAGE
            ;5026 ;
            ;5027 ;////////////////////////////////////////////////////////////////////////////////
            ;5028 ;
            ;5029 ; Got an unexpected interrupt when writing the encoded value out to memory.
            ;5030 ; Call an error
            ;5031 ;
U 10EE, 0000,003D,F580,0800,0084,703C  ;5032  ERROR1[.A]
            ;5033 ;
            ;5034 ;////////////////////////////////////////////////////////////////////////////////
            ;5035 ;
            ;5036 ; ERROR LOGOUT:
            ;5037 ;
            ;5038 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5039 ;  Memory Test Address
            ;5040 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5041 ;       0 = Upper or Lower Present
            ;5042 ;  Unused
            ;5043 ;  Unused
            ;5044 ;  Unused
            ;5045 ;  ID[vector] Register
            ;5046 ;  ID[sbi.err] Register
            ;5047 ;  ID[fault] Register
            ;5048 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;5049 ;       0 = No Interrupt Occurred
            ;5050 ;
            ;5051 ;////////////////////////////////////////////////////////////////////////////////
            ;5052 ;
            ;5053 T.29.S3.DO.SEC:
U 10EF, 001B,0038,0180,09E8,0000,13BD  ;5054  RC[0D]_K[.8]  ; Load RC[0D] with second location address
U 13BD, 001B,0038,0D80,09A0,0000,1218  ;5055  RC[04]_K[.3]  ; Second pattern to be encoded
U 1218, 0000,003D,0180,0800,0000,12F3  ;5056 =0 CALL[ENCODE]  ; Go encode the pattern
U 1219, 0000,003C,0180,0800,0000,121A  ;5057  NOP
U 121A, 0000,003D,0180,0800,0000,1259  ;5058 =0 ERLOOP   ; Loop on error from here
U 121B, 0000,003C,0180,0800,0000,121C  ;5059  NOP
U 121C, 0000,003D,0180,0800,0000,107C  ;5060 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 121D, 0000,003C,0180,0800,0000,121E  ;5061  NOP
U 121E, 081B,0039,0D80,0800,0000,12E7  ;5062 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 121F, 081B,0038,0580,0800,0000,10F0  ;5063  D_K[.1]   ; Load D with 1 for Internally Interleaved
U 10F0, 0000,003D,0180,0800,0000,1263  ;5064 =00 CALL[SELECT.CNTRLR] ; Select Internally Interleaved mode
```

```
            ;5065  .PAGE
            ;5066 ;
            ;5067 ;////////////////////////////////////////////////////////////////////////////
            ;5068 ;
            ;5069 ; Got a misconfiguration error when selecting Non-Interleave mode.  Call an
            ;5070 ; error
            ;5071 ;
            ;5072 T.29.S3.CONFIG.ERR.2:
U 10F1, 0000,003D,0D80,0800,0084,703C  ;5073  ERROR1[.3]
            ;5074 ;
            ;5075 ;////////////////////////////////////////////////////////////////////////////
            ;5076 ;
            ;5077 ; ERROR LOGOUT:
            ;5078 ;
            ;5079 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5080 ;  Memory Test Address
            ;5081 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5082 ;       0 = Upper or Lower Present
            ;5083 ;
            ;5084 ;////////////////////////////////////////////////////////////////////////////
            ;5085 ;
            ;5086 T.29.S3.CONT.SEC.2:
U 10F2, 0010,0038,0180,0968,0200,10F3  ;5087  VA_RC[0D]  ; Point VA to the test location
U 10F3, 0810,0038,0180,0908,0000,13BE  ;5088  D_RC[01]  ; Move encoded test pattern to the D register
U 13BE, 0000,003C,0180,A800,0000,10F4  ;5089  CACHE.P_D[LONG]  ; Write the test location
U 10F4, 0000,003D,0180,0800,0000,1280  ;5090 =00 CHECK.INTERRUPT  ; Any unexpected Interrupts ?
U 10F5, 0000,003C,0180,0800,0000,10F7  ;5091  J/T.29.S3.DO.FIR.READ ; No, go read the first test location
```

```
            ;5092  .PAGE
            ;5093 ;
            ;5094 ;////////////////////////////////////////////////////////////////////////////////
            ;5095 ;
            ;5096 ; Got an unexpected interrupt when writing the second test data to memory.
            ;5097 ; Call an error.
            ;5098 ;
U 10F6, 0000,003D,F580,0800,0084,703C  ;5099  ERROR1[.A]
            ;5100 ;
            ;5101 ;////////////////////////////////////////////////////////////////////////////////
            ;5102 ;
            ;5103 ; ERROR LOGOUT:
            ;5104 ;
            ;5105 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5106 ;  Memory Test Address
            ;5107 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5108 ;       0 = Upper or Lower Present
            ;5109 ;  Unused
            ;5110 ;  Unused
            ;5111 ;  Unused
            ;5112 ;  ID[vector] Register
            ;5113 ;  ID[sbi.err] Register
            ;5114 ;  ID[fault] Register
            ;5115 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;5116 ;       0 = No Interrupt Occurred
            ;5117 ;
            ;5118 ;////////////////////////////////////////////////////////////////////////////////
            ;5119 ;
            ;5120 T.29.S3.DO.FIR.READ:
U 10F7, 0003,003C,0180,09E8,0000,13BF  ;5121  RC[0D]_0  ; Point to first location under test
U 13BF, 0018,0038,0580,09A0,0000,1220  ;5122  RC[04]_K[.1]  ; Set-up pattern counter for expected
            ;5123      ; encoded data from upper controller
U 1220, 0000,003D,0180,0800,0000,1259  ;5124 =0 ERLOOP   ; Loop on error from here
U 1221, 0000,003C,0180,0800,0000,1222  ;5125  NOP
U 1222, 0818,0039,0D80,0800,0000,12E7  ;5126 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 1223, 0818,0038,0980,0800,0000,10F8  ;5127  D_K[.2]   ; Set D reg up to select Upper controller in
            ;5128      ; non interleaved mode
U 10F8, 0000,003D,0180,0800,0000,1263  ;5129 =00 CALL[SELECT.CNTRLR] ; Select upper controller non interleaved mode
```

```
            ;5130  .PAGE
            ;5131 ;
            ;5132 ;////////////////////////////////////////////////////////////////////////////
            ;5133 ;
            ;5134 ; Got misconfiguration error when selecting upper controller non interleaved
            ;5135 ; mode and should have.  Call an error.
            ;5136 ;
U 10F9, 0000,003D,0D80,0800,0084,703C  ;5137  ERROR1[.3]
            ;5138 ;
            ;5139 ;////////////////////////////////////////////////////////////////////////////
            ;5140 ;
            ;5141 ; ERROR LOGOUT:
            ;5142 ;
            ;5143 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5144 ;  Memory Test Address
            ;5145 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5146 ;       0 = Upper or Lower Present
            ;5147 ;
            ;5148 ;////////////////////////////////////////////////////////////////////////////
            ;5149 ;
            ;5150 T.29.S3.NO.MISCONFG:
U 10FA, 0010,0038,0180,0968,0200,10FB  ;5151  VA_RC[0D]  ; Load VA with current test address
U 10FB, 0000,003C,0180,C800,0000,13C0  ;5152  D[LONG]_CACHE.P  ; Read test location 0
U 13C0, 0001,003C,0180,0980,0000,10FC  ;5153  RC[00]_D  ; Save in RC[00] for Decode routine
U 10FC, 0000,003D,0180,0800,0000,1317  ;5154 =00 CALL[DECODE]  ; Go decode the data read
U 10FD, 0000,003C,0180,0800,0000,10FE  ;5155  NOP  ; Return1 if maximum errors
U 10FE, 0000,003C,0180,0800,0000,1224  ;5156  J/T.29.CALL.ERDAT.5 ; Return2 if error but not maximum
U 10FF, 0000,003C,0180,0800,0000,1227  ;5157  J/T.29.S3.NO.DEC.ERR ; No Decode error, continue with test
```

```
            ;5158  .PAGE
            ;5159 ;
            ;5160 ;////////////////////////////////////////////////////////////////////////////////
            ;5161 ;
            ;5162 ; Got a Decode error when reading the test data from the upper controller.
            ;5163 ; Call an error.
            ;5164 ;
            ;5165 =0
            ;5166 T.29.CALL.ERDAT.5:
U 1224, 0000,003D,0180,0800,0000,12DE  ;5167  CALL[ERDAT]  ; Decode error, get the error logout data
U 1225, 0000,003C,0180,0800,0000,1226  ;5168  NOP
U 1226, 0000,003D,6180,0800,0084,703C  ;5169 =0 ERROR1[.F]
            ;5170 ;
            ;5171 ;////////////////////////////////////////////////////////////////////////////////
            ;5172 ;
            ;5173 ; ERROR LOGOUT:
            ;5174 ;
            ;5175 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5176 ;  Memory Test Address
            ;5177 ;  Received Memory Data
            ;5178 ;  Expected Memory Data
            ;5179 ;  Received Memory Data
            ;5180 ;  Expected Memory Data
            ;5181 ;  Received Memory Data
            ;5182 ;  Expected Memory Data
            ;5183 ;  Received Memory Data
            ;5184 ;  Expected Memory Data
            ;5185 ;  Received Memory Data
            ;5186 ;  Expected Memory Data
            ;5187 ;  Received Memory Data
            ;5188 ;  Expected Memory Data
            ;5189 ;  FFFFFFFF
            ;5190 ;
            ;5191 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5192 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;5193 ;        be output on the error then the last longword wrote will be the
            ;5194 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5195 ;
            ;5196 ;////////////////////////////////////////////////////////////////////////////////
            ;5197 ;
            ;5198 T.29.S3.NO.DEC.ERR:
U 1227, 0018,0038,0980,09A0,0000,1228  ;5199  RC[04]_K[.2]  ; Set-up pattern to be encoded
U 1228, 0000,003D,0180,0800,0000,12F3  ;5200 =0 CALL[ENCODE]  ; Encode the data
U 1229, 0018,0038,0D80,09A0,00D0,122A  ;5201  RC[04]_K[.3]  ; Set-up pattern that is expected from lower
            ;5202      ; controller
U 122A, 0000,003D,0180,0800,0000,1259  ;5203 =0 ERLOOP   ; Loop on error from here
U 122B, 0000,003C,0180,0800,0000,122C  ;5204  NOP
U 122C, 0818,0039,0D80,0800,0000,12E7  ;5205 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 122D, 0818,0038,1980,0800,0000,1110  ;5206  D_K[ZERO]  ; Load D with data to select Lower controller
            ;5207      ; in non interleaved mode
U 1110, 0000,003D,0180,0800,0000,1263  ;5208 =00 CALL[SELECT.CNTRLR] ; Select Lower controller in non interleave
```

```
            ;5209  .PAGE
            ;5210 ;
            ;5211 ;////////////////////////////////////////////////////////////////////////////
            ;5212 ;
            ;5213 ; Got a misconfiguration error when selecting lower controller in non
            ;5214 ; interleave mode and should have.  Call an error.
            ;5215 ;
U 1111, 0000,003D,0D80,0800,0084,703C  ;5216  ERROR1[.3]
            ;5217 ;
            ;5218 ;////////////////////////////////////////////////////////////////////////////
            ;5219 ;
            ;5220 ; ERROR LOGOUT:
            ;5221 ;
            ;5222 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5223 ;  Memory Test Address
            ;5224 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5225 ;       0 = Upper or Lower Present
            ;5226 ;
            ;5227 ;////////////////////////////////////////////////////////////////////////////
            ;5228 ;
            ;5229 T.29.S3.READ.LOW:
U 1112, 0010,0038,0180,0968,0200,1113  ;5230  VA_RC[0D]  ; Load VA with test address
U 1113, 0000,003C,0180,C800,0000,13C1  ;5231  D[LONG]_CACHE.P  ; Read the data at the test address
U 13C1, 0001,003C,0180,0980,0000,1114  ;5232  RC[00]_D  ; Save read data in RC[00] for decode
U 1114, 0000,003D,0180,0800,0000,1317  ;5233 =00 CALL[DECODE]  ; Decode the received data
U 1115, 0000,003C,0180,0800,0000,1116  ;5234  NOP  ; Return1 if maximum errors
U 1116, 0000,003C,0180,0800,0000,122E  ;5235  J/T.29.CALL.ERDAT.6 ; Return2 if error but not maximum
U 1117, 0000,003C,0180,0800,0000,1231  ;5236  J/T.29.S3.CONT.1 ; No decode error, continue with test
```

```
            ;5237  .PAGE
            ;5238 ;
            ;5239 ;////////////////////////////////////////////////////////////////////////////////
            ;5240 ;
            ;5241 ; Got a decode error on data from lower controller.  Call an error.
            ;5242 ;
            ;5243 =0
            ;5244 T.29.CALL.ERDAT.6:
U 122E, 0000,003D,0180,0800,0000,12DE  ;5245  CALL[ERDAT]  ; Got decode error, get the error logout data
U 122F, 0000,003C,0180,0800,0000,1230  ;5246  NOP
U 1230, 0000,003D,6180,0800,0084,703C  ;5247 =0 ERROR1[.F]
            ;5248 ;
            ;5249 ;////////////////////////////////////////////////////////////////////////////////
            ;5250 ;
            ;5251 ; ERROR LOGOUT:
            ;5252 ;
            ;5253 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5254 ;  Memory Test Address
            ;5255 ;  Received Memory Data
            ;5256 ;  Expected Memory Data
            ;5257 ;  Received Memory Data
            ;5258 ;  Expected Memory Data
            ;5259 ;  Received Memory Data
            ;5260 ;  Expected Memory Data
            ;5261 ;  Received Memory Data
            ;5262 ;  Expected Memory Data
            ;5263 ;  Received Memory Data
            ;5264 ;  Expected Memory Data
            ;5265 ;  Received Memory Data
            ;5266 ;  Expected Memory Data
            ;5267 ;  FFFFFFFF
            ;5268 ;
            ;5269 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5270 ;       terminated by a xFFFFFFFF.  If there are more errors then can
            ;5271 ;       be output on the error then the last longword wrote will be the
            ;5272 ;       xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5273 ;
            ;5274 ;////////////////////////////////////////////////////////////////////////////////
            ;5275 ;
            ;5276 T.29.S3.CONT.1:
U 1231, 0018,0038,1180,09A0,0000,1232  ;5277  RC[04]_K[.4]  ; Set pattern count to encode data
U 1232, 0000,003D,0180,0800,0000,12F3  ;5278 =0 CALL[ENCODE]  ; Encode the data
U 1233, 0810,0038,0180,0908,0000,13C2  ;5279  D_RC[01]  ; Move encode data to D for write
U 13C2, 0010,0038,0180,0968,0200,13C3  ;5280  VA_RC[0D]  ; Load VA with current test address
U 13C3, 0000,003C,0180,A800,0000,13C4  ;5281  CACHE.P_D[LONG]  ; Write the encoded data
U 13C4, 0018,0038,0580,09A0,0000,13C5  ;5282  RC[04]_K[.1]  ; Set pattern count to encode data that is
            ;5283      ; expected from the upper controller
U 13C5, 0003,003C,0180,09E8,0000,1234  ;5284  RC[0D]_0  ; Point to location to be tested on the
            ;5285      ; upper controller
U 1234, 0000,003D,0180,0800,0000,1259  ;5286 =0 ERLOOP   ; Loop on error from here
U 1235, 0000,003C,0180,0800,0000,1236  ;5287  NOP
U 1236, 0000,003D,0180,0800,0000,107C  ;5288 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1237, 0000,003C,0180,0800,0000,1238  ;5289  NOP
U 1238, 0818,0039,0D80,0800,0000,12E7  ;5290 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 1239, 0818,0038,1180,0800,0000,1118  ;5291  D_K[.4]   ; Set-up to select Internal Interleave
```

```
U 1118, 0000,003D,0180,0800,0000,1263  ;5292 =00 CALL[SELECT.CNTRLR] ; Select internal interleave
U 1119, 0000,003C,0180,0800,0000,111A  ;5293  NOP
U 111A, 0010,0038,0180,0968,0200,123A  ;5294  VA_RC[0D]  ; Load VA with current test address
          ;5295 =
U 123A, 0000,003D,8980,0800,0084,7293  ;5296 =0 SET.TRAP.MASK[.D] ; Enable time out micro traps
U 123B, 0000,003C,0180,C800,0000,13C6  ;5297  D[LONG]_CACHE.P  ; Read the test location
U 13C6, 0001,003C,0180,0980,0000,111C  ;5298  RC[00]_D  ; Save in RC[00] for Decode
U 111C, 0000,003D,0180,0800,0000,128D  ;5299 =00 CHECK.UTRAP  ; Was there a time out micro trap ?
U 111D, 0000,003C,0180,0800,0000,1120  ;5300  J/T.29.S3.CHK.INT ; No, got check for unexpected interrupts
```

```
            ;5301  .PAGE
            ;5302 ;
            ;5303 ;////////////////////////////////////////////////////////////////////////////////
            ;5304 ;
            ;5305 ; Got a time out micro trap when reading the data from the test location.
            ;5306 ; Call an error.
            ;5307 ;
U 111E, 0000,003D,0180,0800,0084,7044  ;5308  ERROR2[.8]
U 111F, 0000,003C,0180,0800,0000,1120  ;5309  NOP
            ;5310 ;
            ;5311 ;////////////////////////////////////////////////////////////////////////////////
            ;5312 ;
            ;5313 ; ERROR LOGOUT:
            ;5314 ;
            ;5315 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5316 ;  Memory Test Address
            ;5317 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5318 ;       0 = Upper or Lower Present
            ;5319 ;  Unused
            ;5320 ;  Unused
            ;5321 ;  Unused
            ;5322 ;  Expected Micro Trap Flags
            ;5323 ;  Received Micro Trap Flags
            ;5324 ;
            ;5325 ;////////////////////////////////////////////////////////////////////////////////
            ;5326 ;
            ;5327 =00
            ;5328 T.29.S3.CHK.INT:
U 1120, 0000,003D,0180,0800,0000,1280  ;5329  CHECK.INTERRUPT  ; Any unexpected interrupts ?
U 1121, 0000,003C,0180,0800,0000,1124  ;5330  J/T.29.S3.CONT.2 ; No, continue with test
```

```
          ;5331  .PAGE
          ;5332  ;
          ;5333  ;////////////////////////////////////////////////////////////////////////////////
          ;5334  ;
          ;5335  ; Got an unexpected interrupt when reading the test data.  Call an error.
          ;5336  ;
U 1122, 0000,003D,F580,0800,0084,703C  ;5337  ERROR1[.A]
          ;5338  =
          ;5339  ;
          ;5340  ;////////////////////////////////////////////////////////////////////////////////
          ;5341  ;
          ;5342  ; ERROR LOGOUT:
          ;5343  ;
          ;5344  ; DATA: I/O Base Address of MS780-E Currently Under Test
          ;5345  ;  Memory Test Address
          ;5346  ;  Controller Present Flag  1 = Upper &  Lower Present
          ;5347  ;       0 = Upper or Lower Present
          ;5348  ;  Unused
          ;5349  ;  Unused
          ;5350  ;  Unused
          ;5351  ;  ID[vector] Register
          ;5352  ;  ID[sbi.err] Register
          ;5353  ;  ID[fault] Register
          ;5354  ;  Interrupt Occurred Flag  1 = Interrupt Occured
          ;5355  ;       0 = No Interrupt Occurred
          ;5356  ;
          ;5357  ;////////////////////////////////////////////////////////////////////////////////
          ;5358  ;
          ;5359  =00
          ;5360 T.29.S3.CONT.2:
U 1124, 0000,003D,0180,0800,0000,1317  ;5361  CALL[DECODE]  ; Decode the received vector
U 1125, 0000,003C,0180,0800,0000,1126  ;5362  NOP  ; Return1 if maximum errors
U 1126, 0000,003C,0180,0800,0000,123C  ;5363  J/T.29.CALL.ERDAT.7 ; Return2 if error but not maximum
U 1127, 0000,003C,0180,0800,0000,123F  ;5364  J/T.29.S3.CONT.3 ; No Decode error, continue with test
```

```
            ;5365  .PAGE
            ;5366 ;
            ;5367 ;////////////////////////////////////////////////////////////////////////////////
            ;5368 ;
            ;5369 ; Got a Decode error on read data.  Call an error
            ;5370 ;
            ;5371 =0
            ;5372 T.29.CALL.ERDAT.7:
U 123C, 0000,003D,0180,0800,0000,12DE  ;5373  CALL[ERDAT]  ; Got Decode error, get the error data
U 123D, 0000,003C,0180,0800,0000,123E  ;5374  NOP
U 123E, 0000,003D,6180,0800,0084,703C  ;5375 =0 ERROR1[.F]
            ;5376 ;
            ;5377 ;////////////////////////////////////////////////////////////////////////////////
            ;5378 ;
            ;5379 ; ERROR LOGOUT:
            ;5380 ;
            ;5381 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5382 ;  Memory Test Address
            ;5383 ;  Received Memory Data
            ;5384 ;  Expected Memory Data
            ;5385 ;  Received Memory Data
            ;5386 ;  Expected Memory Data
            ;5387 ;  Received Memory Data
            ;5388 ;  Expected Memory Data
            ;5389 ;  Received Memory Data
            ;5390 ;  Expected Memory Data
            ;5391 ;  Received Memory Data
            ;5392 ;  Expected Memory Data
            ;5393 ;  Received Memory Data
            ;5394 ;  Expected Memory Data
            ;5395 ;  FFFFFFFF
            ;5396 ;
            ;5397 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5398 ;       terminated by a xFFFFFFFF.  If there are more errors then can
            ;5399 ;       be output on the error then the last longword wrote will be the
            ;5400 ;       xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5401 ;
            ;5402 ;////////////////////////////////////////////////////////////////////////////////
            ;5403 ;
            ;5404 T.29.S3.CONT.3:
U 123F, 0018,0038,1180,09A0,0000,13C7  ;5405  RC[04]_K[.4]  ; Set pattern count to encoded data that is
            ;5406     ; expected from lower controller in Internal
            ;5407     ; interleave mode
U 13C7, 0018,0038,0180,09E8,0000,1240  ;5408  RC[0D]_K[.8]  ; Set up test location - 8
U 1240, 0000,003D,0180,0800,0000,1259  ;5409 =0 ERLOOP   ; Loop on error from here
U 1241, 0000,003C,0180,0800,0000,1242  ;5410  NOP
U 1242, 0000,003D,0180,0800,0000,107C  ;5411 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1243, 0000,003C,0180,0800,0000,1244  ;5412  NOP
U 1244, 0818,0039,0D80,0800,0000,12E7  ;5413 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 1245, 0818,0038,1180,0800,0000,1128  ;5414  D_K[.4]   ; Set-up to select Internal Interleave
U 1128, 0000,003D,0180,0800,0000,1263  ;5415 =00 CALL[SELECT.CNTRLR] ; Select Internal interleave mode
U 1129, 0000,003C,0180,0800,0000,112A  ;5416  NOP
U 112A, 0010,0038,0180,0968,0200,1246  ;5417  VA_RC[0D]  ; Load VA with current test address
            ;5418 =
U 1246, 0000,003D,8980,0800,0084,7293  ;5419 =0 SET.TRAP.MASK[.D] ; Enable time out micro traps
```

```
U 1247, 0000,003C,0180,C800,0000,13C8  ;5420  D[LONG]_CACHE.P  ; Read the data from the test address
U 13C8, 0001,003C,0180,0980,0000,112C  ;5421  RC[00]_D  ; Save read data in RC[00] for Decode
U 112C, 0000,003D,0180,0800,0000,128D  ;5422 =00 CHECK.UTRAP  ; Was there a time out micro trap ?
U 112D, 0000,003C,0180,0800,0000,1130  ;5423  J/T.29.S3.CHK.INT.2 ; No, got check for unexpected interrupts
```

```
          ;5424  .PAGE
          ;5425 ;
          ;5426 ;////////////////////////////////////////////////////////////////////////////////
          ;5427 ;
          ;5428 ; Got a time out micro trap when reading the data from the test location.
          ;5429 ; Call an error.
          ;5430 ;
U 112E, 0000,003D,0180,0800,0084,7044  ;5431  ERROR2[.8]
U 112F, 0000,003C,0180,0800,0000,1130  ;5432  NOP
          ;5433 ;
          ;5434 ;////////////////////////////////////////////////////////////////////////////////
          ;5435 ;
          ;5436 ; ERROR LOGOUT:
          ;5437 ;
          ;5438 ; DATA: I/O Base Address of MS780-E Currently Under Test
          ;5439 ;  Memory Test Address
          ;5440 ;  Controller Present Flag  1 = Upper &  Lower Present
          ;5441 ;       0 = Upper or Lower Present
          ;5442 ;  Unused
          ;5443 ;  Unused
          ;5444 ;  Unused
          ;5445 ;  Expected Micro Trap Flags
          ;5446 ;  Received Micro Trap Flags
          ;5447 ;
          ;5448 ;////////////////////////////////////////////////////////////////////////////////
          ;5449 ;
          ;5450 =00
          ;5451 T.29.S3.CHK.INT.2:
U 1130, 0000,003D,0180,0800,0000,1280  ;5452  CHECK.INTERRUPT  ; Any unexpected interrupts ?
U 1131, 0000,003C,0180,0800,0000,1134  ;5453  J/T.29.S3.CONT.4 ; No, continue with test
```

```
            ;5454  .PAGE
            ;5455 ;
            ;5456 ;////////////////////////////////////////////////////////////////////////////////
            ;5457 ;
            ;5458 ; Got an unexpected interrupt when reading the test data.  Call an error.
            ;5459 ;
U 1132, 0000,003D,F580,0800,0084,703C  ;5460  ERROR1[.A]
U 1133, 0000,003C,0180,0800,0000,1134  ;5461  NOP
            ;5462 ;
            ;5463 ;////////////////////////////////////////////////////////////////////////////////
            ;5464 ;
            ;5465 ; ERROR LOGOUT:
            ;5466 ;
            ;5467 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5468 ;  Memory Test Address
            ;5469 ;  Controller Present Flag  1 = Upper &  Lower Present
            ;5470 ;       0 = Upper or Lower Present
            ;5471 ;  Unused
            ;5472 ;  Unused
            ;5473 ;  Unused
            ;5474 ;  ID[vector] Register
            ;5475 ;  ID[sbi.err] Register
            ;5476 ;  ID[fault] Register
            ;5477 ;  Interrupt Occurred Flag  1 = Interrupt Occured
            ;5478 ;       0 = No Interrupt Occurred
            ;5479 ;
            ;5480 ;////////////////////////////////////////////////////////////////////////////////
            ;5481 ;
            ;5482 =00
            ;5483 T.29.S3.CONT.4:
U 1134, 0000,003D,0180,0800,0000,1317  ;5484  CALL[DECODE]  ; Decode the received vector
U 1135, 0000,003C,0180,0800,0000,1136  ;5485  NOP    ; Return1 if maximum errors
U 1136, 0000,003C,0180,0800,0000,1248  ;5486  J/T.29.CALL.ERDAT.8 ; Return2 if error but not maximum
U 1137, 0000,003C,0180,0800,0000,1138  ;5487  J/T.29.CHECK.NEXT ; No Decode error, continue with test
```

```
            ;5488  .PAGE
            ;5489 ;
            ;5490 ;////////////////////////////////////////////////////////////////////////////////
            ;5491 ;
            ;5492 ; Got a Decode error on read data.  Call an error
            ;5493 ;
            ;5494 =0
            ;5495 T.29.CALL.ERDAT.8:
U 1248, 0000,003D,0180,0800,0000,12DE  ;5496  CALL[ERDAT]  ; Got Decode error, get the error data
U 1249, 0000,003C,0180,0800,0000,124A  ;5497  NOP
U 124A, 0000,003D,6180,0800,0084,703C  ;5498 =0 ERROR1[.F]
U 124B, 0000,003C,0180,0800,0000,1138  ;5499  NOP
            ;5500 ;
            ;5501 ;////////////////////////////////////////////////////////////////////////////////
            ;5502 ;
            ;5503 ; ERROR LOGOUT:
            ;5504 ;
            ;5505 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5506 ;  Memory Test Address
            ;5507 ;  Received Memory Data
            ;5508 ;  Expected Memory Data
            ;5509 ;  Received Memory Data
            ;5510 ;  Expected Memory Data
            ;5511 ;  Received Memory Data
            ;5512 ;  Expected Memory Data
            ;5513 ;  Received Memory Data
            ;5514 ;  Expected Memory Data
            ;5515 ;  Received Memory Data
            ;5516 ;  Expected Memory Data
            ;5517 ;  Received Memory Data
            ;5518 ;  Expected Memory Data
            ;5519 ;  FFFFFFFF
            ;5520 ;
            ;5521 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5522 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;5523 ;        be output on the error then the last longword wrote will be the
            ;5524 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5525 ;
            ;5526 ;////////////////////////////////////////////////////////////////////////////////
            ;5527 ;
            ;5528 =00
            ;5529 T.29.CHECK.NEXT:
U 1138, 0000,003D,0180,0800,0000,12C9  ;5530  CALL[ENABLE.NEXT.MS780E]
            ;5531      ; Are there any more MS780-E's to test ?
U 1139, 0000,003C,0180,0800,0000,1010  ;5532  J/RESTART.TEST.29 ; Yes, restart this test
U 113A, 0000,003C,0180,0800,0000,113B  ;5533  NOP    ; No, exit this test
            ;5534  ;
            ;5535  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;5536  ;
            ;5537 T.29.EXIT:
U 113B, 0000,003C,0180,0800,0000,124C  ;5538  NOP
            ;5539
```

```
            ;5540  .PAGE "TEST 1B8 RESERVED"
            ;5541 =0
U 124C, 0000,003D,0180,0800,0000,109B  ;5542 T1B8: NEWTST
U 124D, 0000,003C,0180,0800,0000,13C9  ;5543  NOP
            ;5544
```

```
            ;5545  .PAGE "DUMMY NEWTST"
U 13C9, 0000,003D,0180,0800,0000,109B  ;5546 DUM: NEWTST
            ;5547
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1931: 1910 1919= 1920= 2555= 2556= 2557= 2560=
U 1008  2318= 2319= 2320= 2322= 2271= 2272= 2273= 2274=
U 1010  3864= 3865= 3867= 3869= 3871= 1911 1921= 1922=
U 1018  2684= 2685= 2686= 2687= 2688= 2689= 2690= 2691=
U 1020  2366= 2367= 2368= 2370= 2924= 2925= 2932= 2934=
U 1028  1974= 1976= 1912 2964= 1987= 1988= 1913 2965=
U 1030  3727= 3728= 3729= 3731= 3733= 3734= 3737= 3740=
U 1038  3745= 3747= 3749= 3752= 2048= 2049= 3050= 3051=
U 1040  3842= 3843= 3844= 1915 2094= 2095= 1916 3481=
U 1048  3947= 3956= 3974= 3975= 2219= 2220= 2030: 3485=
U 1050  3977= 3978= 3986= 1923 2401= 2406= 1924 3576=
U 1058  2411= 2416= 2076: 1925 2421= 2422= 2110: 3579=
U 1060  3167= 3168= 3169= 3170= 3171= 3172= 3173= 3174=
U 1068  3175= 3176= 3177= 3178= 3179= 3180= 3181= 3184=
U 1070  4010= 4012= 4013= 1926 4024= 4033= 4034= 1927
U 1078  4054= 4055= 4063= 4064= 2467= 2468= 3467= 3468=
U 1080  4084= 4085= 4093= 4094= 4117= 4118= 4119= 4120=
U 1088  4172= 4173= 4174= 1928 2595= 2596= 3563= 3564=
U 1090  4216= 4217= 4225= 4247= 2663= 2664= 1929 3628=
U 1098  4254= 4262= 4276= 1963 2672= 2673= 1964 3629=
U 10A0  4281= 4282= 4290= 4291= 4311= 4312= 4320= 4321=
U 10A8  4344= 4345= 4346= 4347= 4396= 4397= 4407= 4421=
U 10B0  4475= 4484= 4498= 4499= 4501= 4502= 4510= 4511=
U 10B8  4534= 4536= 4537= 1965 4547= 4556= 4557= 1966
U 10C0  4577= 4578= 4586= 4587= 4607= 4608= 4616= 4617=
U 10C8  4640= 4641= 4642= 4643= 4695= 4696= 4697= 1967
U 10D0  4739= 4740= 4748= 4770= 4777= 4785= 4799= 1968
U 10D8  4804= 4805= 4813= 4814= 4834= 4835= 4843= 4844=
U 10E0  4867= 4868= 4869= 4870= 4919= 4920= 4930= 4944=
U 10E8  4997= 5006= 5020= 5021= 5023= 5024= 5032= 5054=
U 10F0  5064= 5073= 5087= 5088= 5090= 5091= 5099= 5121=
U 10F8  5129= 5137= 5151= 5152= 5154= 5155= 5156= 5157=
U 1100  1896: 1897: 1898: 1899: 1900: 1901: 1902: 1903:
U 1108  1904: 1969 2674= 2676= 1905: 1906: 1907: 1908:
U 1110  5208= 5216= 5230= 5231= 5233= 5234= 5235= 5236=
U 1118  5292= 5293= 5294= 1970 5299= 5300= 5308= 5309=
U 1120  5329= 5330= 5337= 1971 5361= 5362= 5363= 5364=
U 1128  5415= 5416= 5417= 1972 5422= 5423= 5431= 5432=
U 1130  5452= 5453= 5460= 5461= 5484= 5485= 5486= 5487=
U 1138  5530= 5532= 5533= 5538= 2679= 2680= 2721= 2722=
U 1140  2723= 2724= 2731= 2732= 2738= 2740= 2743= 2744=
U 1148  2763= 2764= 2766= 2767= 2858= 2859= 2862= 2863=
U 1150  2874= 2875= 2878= 2879= 3118= 3120= 3123= 3125=
U 1158  3126= 3128= 3240= 3241= 3245= 3246= 3260= 3262=
U 1160  3320= 3321= 3325= 3326= 3328= 3329= 3340= 3341=
U 1168  3353= 3354= 3417= 3419= 3426= 3427= 3434= 3435=
U 1170  3442= 3443= 3448= 3451= 3475= 3477= 3490= 3494=
U 1178  3497= 3498= 3504= 3507= 3529= 3531= 3538= 3539=
U 1180  3544= 3547= 3571= 3573= 3583= 3584= 3586= 3589=
U 1188  3602= 3604= 3613= 3616= 3621= 3622= 3645= 3647=
U 1190  3668= 3670= 3840= 3841= 3910= 3912= 3930= 3931=
U 1198  3934= 3935= 3936= 3937= 3938= 3939= 3940= 3941=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4017= 4018= 4019= 4020= 4021= 4022= 4050= 4051=
U 11A8  4131= 4132= 4133= 4134= 4165= 4166= 4168= 4169=
U 11B0  4170= 4171= 4185= 4204= 4206= 4207= 4208= 4209=
U 11B8  4210= 4211= 4212= 4213= 4248= 4249= 4250= 4251=
U 11C0  4252= 4253= 4278= 4279= 4356= 4357= 4358= 4388=
U 11C8  4390= 4391= 4392= 4393= 4394= 4395= 4430= 4431=
U 11D0  4439= 4440= 4461= 4462= 4465= 4466= 4467= 4468=
U 11D8  4469= 4470= 4471= 4472= 4541= 4542= 4543= 4544=
U 11E0  4545= 4546= 4573= 4574= 4654= 4655= 4656= 4657=
U 11E8  4688= 4689= 4691= 4692= 4693= 4694= 4708= 4727=
U 11F0  4729= 4730= 4731= 4732= 4733= 4734= 4735= 4736=
U 11F8  4771= 4772= 4773= 4774= 4775= 4776= 4801= 4802=
U 1200  4879= 4880= 4881= 4911= 4913= 4914= 4915= 4916=
U 1208  4917= 4918= 4953= 4954= 4962= 4963= 4984= 4985=
U 1210  4988= 4989= 4990= 4991= 4992= 4993= 4994= 4995=
U 1218  5056= 5057= 5058= 5059= 5060= 5061= 5062= 5063=
U 1220  5124= 5125= 5126= 5127= 5167= 5168= 5169= 5199=
U 1228  5200= 5201= 5203= 5204= 5205= 5206= 5245= 5246=
U 1230  5247= 5277= 5278= 5279= 5286= 5287= 5288= 5289=
U 1238  5290= 5291= 5296= 5297= 5373= 5374= 5375= 5405=
U 1240  5409= 5410= 5411= 5412= 5413= 5414= 5419= 5420=
U 1248  5496= 5497= 5498= 5499= 5542= 5543= 1973 1977
U 1250  1978 1979 1980 1981 1982 1983 1984 1985
U 1258  1986 2005 2028 2029 2074 2075 2158 2159
U 1260  2160 2179 2181 2208 2209 2210 2211 2212
U 1268  2213 2214 2215 2217 2218 2265 2266 2267
U 1270  2269 2270 2396 2445 2469 2470 2471 2472
U 1278  2491 2492 2493 2494 2513 2514 2515 2516
U 1280  2544 2546 2547 2548 2550 2551 2552 2553
U 1288  2554 2561 2562 2563 2564 2588 2589 2590
U 1290  2591 2593 2594 2623 2624 2625 2665 2666
U 1298  2667 2668 2677 2678 2682 2683 2696 2697
U 12A0  2699 2700 2701 2703 2708 2709 2710 2712
U 12A8  2717 2719 2136: 2720 2728 2729 2733 2734
U 12B0  2735 2736 2737 2745 2746 2747 2749 2750
U 12B8  2751 2752 2753 2754 2759 2761 2762 2765
U 12C0  2790 2791 2792 2793 2822 2823 2825 2826
U 12C8  2827 2854 2856 2857 2861 2865 2866 2867
U 12D0  2868 2870 2871 2872 2873 2876 2877 2918
U 12D8  2919 2920 2921 2922 2936 2938 2956 2957
U 12E0  2958 2959 2960 2961 2962 2963 2967 2991
U 12E8  2992 2993 2994 2995 2996 3044 3045 3046
U 12F0  3047 3048 3049 3110 3112 3113 3114 3116
U 12F8  3117 3162 3165 3233 3234 3236 3238 3239
U 1300  3248 3250 3251 3252 3253 3255 3257 3312
U 1308  3313 3314 3316 3318 3319 3331 3332 3336
U 1310  3338 3342 3344 3345 3346 3348 3350 3395
U 1318  3396 3397 3398 3399 3401 3402 3403 3404
U 1320  3405 3406 3407 3409 3410 3412 3413 3415
U 1328  3416 3420 3421 3422 3424 3425 3428 3429
U 1330  3432 3433 3436 3437 3438 3440 3441 3445
U 1338  3446 3453 3454 3456 3459 3460 3461 3463
U 1340  3464 3465 3466 3471 3472 3480 3487 3489
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1062
ESKAR48.MCR                    MICRO2  1P(00)    26-JUL-85  17:18:58     Page  15

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3495 3499 3500 3501 3502 3503 3508 3510
U 1350  3511 3513 3514 3515 3516 3517 3518 3519
U 1358  3521 3522 3524 3525 3527 3528 3532 3533
U 1360  3534 3536 3537 3541 3542 3549 3550 3552
U 1368  3555 3556 3557 3559 3560 3561 3562 3567
U 1370  3568 3575 3581 3582 3585 3590 3591 3593
U 1378  3594 3596 3598 3600 3606 3608 3611 3619
U 1380  3620 3625 3627 3640 3642 3643 3650 3653
U 1388  3656 3657 3659 3660 3661 3662 3663 3665
U 1390  3667 3882 3902 3915 3921 3932 3933 3945
U 1398  3976 4023 4052 4053 4205 4214 4215 4280
U 13A0  4389 4422 4423 4424 4425 4426 4428 4429
U 13A8  4463 4464 4473 4500 4575 4576 4728 4737
U 13B0  4738 4803 4912 4945 4946 4947 4948 4949
U 13B8  4951 4952 4986 4987 5022 5055 5089 5122
U 13C0  5153 5232 5280 5281 5282 5284 5298 5408
U 13C8  5421 5546
U 13D0 - 13EF Unused
U 13F0  2120: 2121: 2122: 2123: 2124: 2125: 2126: 2127:
U 13F8  2128: 2129: 2130: 2131: 2132: 2133: 2134: 2135:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR48	986
Used	986
Remaining	

Total microwords used in memory U: 986
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR48.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 1065
ESKAR49.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1852 Micro Trap Handler and LSI-11 Support Routines
; 1853   Micro Trap Handler
; 1933 Micro Monitor Interface Routines
; 1934   Newtest Routine
; 1990   Error Loop Routine
; 2007   Error 1 Routine
; 2032   ERROR1 WITH PARAMETER
; 2053   Error 2 Routine
; 2079   ERROR 2 WITH PARAMETER
; 2098   Micro-Monitor Call
; 2112   Reserved Control Store
; 2139   Set Argument Count
; 2163   Increment Subtest Number
; 2184 Diagnostic Specific Subroutines
; 2185   Select Controller
; 2223   64K OR 256K CHIPS
; 2278   START TEST
; 2488   SELECT LOWER CONTROLLER
; 2536   SELECT UPPER CONTROLLER
; 2585   SBI Unjam Routine
; 2636   RESET SUBTEST NUMBER
; 2658   Initialize the SBI
; 2686   Enable Cache
; 2708   Disable Cache Routine
; 2730   Check for Interrupt Routine
; 2778   Set Trap Mask
; 2807   FIND MS780-E MEMORY
; 2949   Generate IO Address
; 2975   Set Starting Address
; 3009   ENABLE NEXT MS780-E
; 3061   GET MEMORY SIZE
; 3119   ERDAT ROUTINE
; 3148   Set Diagnostic Mode Routine
; 3177   Check for Odd or Even TR
; 3233   BOARD ADDRESS GENERATOR
; 3316   BANK ADDRESS GENEREATOR
; 3426   REED MULLER ENCODING ROUTINE
; 3502   REED-MULLER VECTOR GENERATION ROUTINE
; 3555   FIRST ORDER REED-MULLER ENCODE
; 3636   SECOND ORDER REED-MULLER ENCODING ROUTINE
; 3727   REED-MULLER DECODE ROUTINE
; 4043   FIND CONTROLLERS
; 4125   DELAY
; 4165 TEST 1B9 BOARD/BANK SELECT ADDRESSING
; 5152 TEST 1BA RESERVED
; 5157 DUMMY NEWTST

```
:1  .NOLIST
:1804  .LIST
:1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR49
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; MODIFIED BY:
;1836 ;
;1837 ;  1.1 March 10, 1983  Paul Hakala
;1838 ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839 ;   the correct memory address when 256k ram chips are
;1840 ;   present.
;1841 ;
;1842 ;   March 10, 1983  Paul Hakala
;1843 ;   Removed  the LOAD.TEMP.REG routine that wasn't
;1844 ;   needed in overlay.
;1845 ;
;1846 ;
;1847 ;
;1848 ;*****************************************************************************
;1849 ;
```

```
;1850  .PAGE
;1851
;1852  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1853  .TOC "   Micro Trap Handler"
;1854 ;++
;1855 ; FUNCTIONAL DESCRIPTION:
;1856 ;
;1857 ; This routine is responsible for 'catching' micro-traps
;1858 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1859 ; contains the Trap-Expected Mask. If the specified bit is set in
;1860 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1861 ; an error call is made.
;1862 ;
;1863 ; INPUT PARAMETERS:
;1864 ;
;1865 ; R[0F] - Expected Trap Mask
;1866 ;  Trap Code Function
;1867 ;  ---------      --------
;1868 ;       0  System Init
;1869 ;       1  Data Unaligned
;1870 ;       2  Cross Page
;1871 ;       3  TB Modify
;1872 ;       4  TB Protection
;1873 ;       6  TB Miss
;1874 ;       7  TB Parity
;1875 ;       8  Cache Parity
;1876 ;       9  Reserved Floating Operand
;1877 ;       C  Read Data Substitute
;1878 ;       D  Time out
;1879 ;       F  Control Store Parity Error
;1880 ;      10  Odd Address
;1881 ;
;1882 ; OUTPUT PARAMETERS:
;1883 ;
;1884 ; R[0F] - Mask bit is cleared.
;1885 ;
;1886 ; DATA: Micro PC of Micro Trap
;1887 ;  Trap Code (See Above)
;1888 ;  Fault Status Register
;1889 ;  SBI Error Register
;1890 ;  Timeout Address Register
;1891 ;  Maintenance Register
;1892 ;  Cache Parity Register
;1893 ;  TB Error Register 1
;1894 ;  TB Error Register 0
;1895 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1896 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1897 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1898 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1899 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1900 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1901 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1902 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1903 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1904 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C. 0000.003C.8580.0800.0084.7001  ;1905 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D. 0000.003C.8980.0800.0084.7001  ;1906 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E. 0000.003C.6580.0800.0084.7001  ;1907 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F. 0000.003C.6180.0800.0084.7003  ;1908 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1909
U 1001. 0000.003C.81F0.2C00.0000.103A  ;1910 trph0: q_id[ustack]  ; Get upc of trap
U 103A. 0C00.003C.01E0.0800.0000.103E  ;1911  q_d,d_q   ; Setup to put it back on stack
U 103E. 0C00.003C.81E0.3C00.0000.1046  ;1912  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1046. 0000.003C.01C0.0A78.0000.1053  ;1913  q_r[0f]   ; Get the Expected Trap Mask
          ;1914  alu_q.andnot.mask,
U 1053. 0001.2024.0180.0800.0010.1056  ;1915   clk.ubcc ; Was trap expected?
U 1056. 0000.013C.0180.0800.0000.1002  ;1916  z?
          ;1917 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1918       alu_q.and.mask, ; ...
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1919       return0  ; ...
U 1003. 0018.0038.1D80.09E8.0000.1038  ;1920 trph1: rc[0d]_sc  ; Output the Trap Code
U 1038. 0000.003D.D980.0800.0084.71B7  ;1921 =0 setrcf[.9]  ; Set output count
U 1039. 0000.003C.49F0.2C00.0000.105B  ;1922  q_id[tber0]  ; Output all the error registers
U 105B. 0001.203C.4DF0.2DB0.0000.1063  ;1923  rc[6]_q,q_id[tber1] ; ...
U 1063. 0001.203C.65F0.2DB8.0000.106B  ;1924  rc[7]_q,q_id[sbi.err] ; ...
U 106B. 0001.203C.6DF0.2DD8.0000.1073  ;1925  rc[0b]_q,q_id[fault] ; ...
U 1073. 0001.203C.75F0.2DE0.0000.1077  ;1926  rc[0c]_q,q_id[maint] ; ...
U 1077. 0001.203C.79F0.2DC8.0000.107B  ;1927  rc[9]_q,q_id[parity] ; ...
U 107B. 0001.203C.69F0.2DC0.0000.1083  ;1928  rc[8]_q,q_id[time.addr] ; ...
U 1083. 0001.203C.81F0.2DD0.0000.1000  ;1929  rc[0a]_q,q_id[ustack] ; ...
          ;1930 1000: ERROR1[.C],  ;
U 1000. 0001.203D.8580.09F0.0084.704C  ;1931   rc[0e]_q ; Report the error
```

```
          ;1932  .PAGE
          ;1933  .TOC " Micro Monitor Interface Routines"
          ;1934  .TOC "   Newtest Routine"
          ;1935 ;++
          ;1936 ; FUNCTIONAL DESCRIPTION:
          ;1937 ;
          ;1938 ; This routine Initializes the following registers:
          ;1939 ;  PSL    to 1F
          ;1940 ;  CES    to 0
          ;1941 ;  ACC.CS    to disable accellerator
          ;1942 ;  SIR    to 0
          ;1943 ;  CLK.CS    to stop interval timer
          ;1944 ;  TBER0    to disable the TB
          ;1945 ;  SBI.MAINT to     0
          ;1946 ;  SBI.ERR    to 0
          ;1947 ;  FAULT    to 0
          ;1948 ;  COMP    to 0
          ;1949 ;  RC[0E]    to  0
          ;1950 ;  R[0F]    to 0
          ;1951 ; It also performs an SBI UNJAM and disables the CACHE.
          ;1952 ; A SCOPE call is then made to the Monitor.
          ;1953 ;
          ;1954 ; CALLING CONSTRAINT:
          ;1955 ;
          ;1956 ; =0
          ;1957 ;
          ;1958 ; SIDE EFFECTS:
          ;1959 ;
          ;1960 ; D Reg is destroyed
          ;1961 ; SC is destroyed
          ;1962 ;--
U 1087, 0000.003C.65F8.0800.0084.708B  ;1963 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 108B, 0818.0038.8D80.0800.0000.10A3  ;1964  d_k[.1f]  ; D=1F
U 10A3, 0D00.003C.0180.0800.0000.10A7  ;1965  d_dai.sc  ; D=001F0000
U 10A7, 0000.003C.3D80.3C00.0000.10B7  ;1966  id[psl]_d  ; psl = 001F0000
U 10B7, 0818.0038.0580.0800.0000.10BB  ;1967  d_k[.1]  ; Clear the CES register
U 10BB, 0D00.003C.0180.0800.0000.10C6  ;1968  d_dai.sc  ; D = 00010000
U 10C6, 0F00.003C.3180.3C00.0000.10CE  ;1969  id[ces]_d,d_0  ; ces = 00010000
U 10CE, 0000.003C.5D80.3C00.0000.1109  ;1970  id[acc.cs]_d  ; acc.cs = 0
U 1109, 0000.003C.3980.3C00.0000.11A6  ;1971  id[sir]_d  ; sir = 0
U 11A6, 0000.003C.2980.3C00.0000.11A7  ;1972  id[clk.cs]_d  ; clk.cs = 0
U 11A7, 0000.003C.4980.3C00.0000.103C  ;1973  id[tber0]_d  ; tber0 = 0
U 103C, 0000.003D.0180.0800.0000.11D2  ;1974 =0 call[sbi.unjam]  ; Unjam the SBI
          ;1975  intrpt.strobe,  ; Strobe interrupts
U 103D, 0818.0038.C180.0800.4000.11A8  ;1976      d_k[.ffff]  ; Setup to clear SBI error register and
U 11A8, 0801.0028.6580.3C00.0000.11A9  ;1977  id[sbi.err]_d,d_not.d ; and fault register
U 11A9, 0F00.003C.6D80.3C00.0000.11AA  ;1978  id[fault]_d,d_0  ; ...
U 11AA, 0000.003C.6D80.3C00.0000.11AB  ;1979  id[fault]_d  ; fault = 0
U 11AB, 0000.003C.7580.3C00.0000.11AC  ;1980  id[maint]_d  ; MAINT = 0
U 11AC, 0000.003C.6580.3C00.0000.11AD  ;1981  id[sbi.err]_d  ; sbi error reg = 0
U 11AD, 0000.003C.7180.3C00.0000.11AE  ;1982  id[comp]_d  ; comp reg = 0
U 11AE, 0000.003C.E580.3C00.0000.11AF  ;1983  ID[39]_d  ; Clear code for subroutine START.TEST
U 11AF, 0001.003C.0180.09F0.0000.11B0  ;1984  rc[0e]_d  ;
U 11B0, 0001.003C.0180.0AF8.0000.11B1  ;1985  r[0f]_d  ; Clear expected trap mask
U 11B1, 0001.003C.0180.09F8.0000.1044  ;1986  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1044, 0000,003D,0180,0800,0000,11DC  ;1987 =0 call[disable.cache] ; Disable the cache
U 1045, 0F03,003C,0180,09E8,0000,105E  ;1988  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1989  .PAGE
            ;1990  .TOC "   Error Loop Routine"
            ;1991 ;++
            ;1992 ; FUNCTIONAL DESCRIPTION:
            ;1993 ;
            ;1994 ; This routine is called with the "ERLOOP" macro. The
            ;1995 ; routine calls the Micro Monitor to setup an error loop.
            ;1996 ;
            ;1997 ; CALLING CONSTRAINT:
            ;1998 ;
            ;1999 ; =0
            ;2000 ;
            ;2001 ; SIDE EFFECTS:
            ;2002 ;
            ;2003 ; D Reg - Destroyed
            ;2004 ;--
U 11B2, 0818,0038,0980,0800,0000,105E  ;2005 ERLOOP: d_k[.2],j/callcon
```

```
            ;2006  .PAGE
            ;2007  .TOC "   Error 1 Routine"
            ;2008 ;++
            ;2009 ; FUNCTIONAL DESCRIPTION:
            ;2010 ;
            ;2011 ; This routine is used to report an error to the user. This
            ;2012 ; routine is used when you do not wish to continue in the
            ;2013 ; same test after the call to the Monitor.
            ;2014 ;
            ;2015 ; CALLING CONSTRAINT:
            ;2016 ;
            ;2017 ; None
            ;2018 ;
            ;2019 ; INPUTS:
            ;2020 ;
            ;2021 ; rc[0f]<15:8> - Contain the subtest number
            ;2022 ;
            ;2023 ; OUTPUTS:
            ;2024 ;
            ;2025 ; D<15:8> - Subtest number
            ;2026 ; D<7:0> - Error 1 Monitor Code
            ;2027 ;--
U 11B3, 0810.0038.0180.0978.0000.11B4  ;2028 ERROR1: d_rc[0f]  ; Get the subtest number
U 11B4, 0819.0034.4D80.0800.0000.104E  ;2029  d_d.and.k[.ff00] ; ...
U 104E, 0819.0030.1180.0800.0000.105E  ;2030 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2031  .PAGE
            ;2032  .TOC "   ERROR1 WITH PARAMETER"
            ;2033 ;++
            ;2034 ; FUNCTIONAL DESCRIPTION:
            ;2035 ;
            ;2036 ; This subroutine takes  as parameter the number of arguments
            ;2037 ; to be printed to the console in the case of an error logout.
            ;2038 ;
            ;2039 ; CALLING CONSTRAINT:
            ;2040 ;
            ;2041 ; =0
            ;2042 ;INPUTS:
            ;2043 ;
            ;2044 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2045 ;--
            ;2046 =0
            ;2047 ERR1.ARG:
U 104C, 0000,003D,0180,0800,0000,11B7  ;2048  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 104D, 0000,003C,0180,0800,0000,11B3  ;2049  J/ERROR1  ; Go to ERROR1
            ;2050 ;
            ;2051 ;
```

```
            ;2052  .PAGE
            ;2053  .TOC "   Error 2 Routine"
            ;2054 ;++
            ;2055 ; FUNCTIONAL DESCRIPTION:
            ;2056 ;
            ;2057 ; This routine is used  to report an error to the user. This
            ;2058 ; routine is used when you wish to continue in the same test
            ;2059 ;       after the call to the Monitor.
            ;2060 ;
            ;2061 ; CALLING CONSTRAINT:
            ;2062 ;
            ;2063 ; =0
            ;2064 ;
            ;2065 ; INPUTS:
            ;2066 ;
            ;2067 ; rc[0f]<15:8> - Contain the subtest number
            ;2068 ;
            ;2069 ; OUTPUTS:
            ;2070 ;
            ;2071 ; D<15:8> - Subtest number
            ;2072 ; D<7:0> - Error 2 Monitor Code
            ;2073 ;--
U 11B5, 0810,0038,0180,0978,0000,11B6  ;2074 ERROR2: d_rc[0f]  ; Get the subtest number
U 11B6, 0819,0034,4D80,0800,0000,105A  ;2075  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2076 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2077 ;
```

```
            ;2078  .PAGE
            ;2079  .TOC "   ERROR 2 WITH PARAMETER"
            ;2080 ;**
            ;2081 ; FUNCTIONAL DESCRIPTION:
            ;2082 ;
            ;2083 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2084 ; except that in this case after setting the number of arguments
            ;2085 ; too the console, control is transferred to routine ERROR2.
            ;2086 ;
            ;2087 ; INPUTS:
            ;2088 ;
            ;2089 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2090 ;
            ;2091 ;--
            ;2092 =0
            ;2093 ERR2.ARG:
U 1054, 0000,003D,0180,0800,0000,11B7  ;2094  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1055, 0000,003C,0180,0800,0000,11B5  ;2095  J/ERROR2  ; Go to ERROR2
            ;2096 ;
```

```
            ;2097  .PAGE
            ;2098  .TOC "   Micro-Monitor Call"
            ;2099 ;*
            ;2100 ; FUNCTIONAL DESCRIPTION:
            ;2101 ;
            ;2102 ; This micro word calls the micro monitor.
            ;2103 ;
            ;2104 ; EXPLICIT INPUTS:
            ;2105 ;
            ;2106 ; D reg must contain valid monitor code.
            ;2107 ;--
            ;2108 105E:
            ;2109 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2110  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2111  .PAGE
            ;2112  .TOC "   Reserved Control Store"
            ;2113 ;**
            ;2114 ; FUNCTIONAL DESCRIPTION:
            ;2115 ;
            ;2116 ; The following 16 locations must be reserved so the monitor
            ;2117 ; can use them to perform various functions that require
            ;2118 ; the execution of micro-code.
            ;2119 ;--
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2120 13F0: nop
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2121 13F1: nop
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2122 13F2: nop
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2123 13F3: nop
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2124 13F4: nop
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2125 13F5: nop
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2126 13F6: nop
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2127 13F7: nop
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2128 13F8: nop
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2129 13F9: nop
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2130 13FA: nop
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2131 13FB: nop
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2132 13FC: nop
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2133 13FD: nop
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2134 13FE: nop
U 13FF, 0000,003C,0180,0800,0000,12AA  ;2135 13FF: nop
U 12AA, 0000,003C,0180,0800,0000,11B7  ;2136 12AA: nop ; This location is permanently blasted in the FPLA
            ;2137   ; to cause a micro ECO to location 12AA.
```

```
            ;2138  .PAGE
            ;2139  .TOC "   Set Argument Count"
            ;2140 ;++
            ;2141 ; FUNCTIONAL DESCRIPTION:
            ;2142 ;
            ;2143 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2144 ; to the console.
            ;2145 ;
            ;2146 ; CALLING CONSTRAINT:
            ;2147 ;
            ;2148 ; =0
            ;2149 ;
            ;2150 ; INPUTS:
            ;2151 ;
            ;2152 ; SC - Contains new value of argument count
            ;2153 ;
            ;2154 ; SIDE EFFECTS:
            ;2155 ;
            ;2156 ; Q is destroyed
            ;2157 ;--
U 11B7, 0010,0038,01C0,0978,0000,11B8  ;2158 SETRCF: q_rc[0f]  ; Get the argument count
U 11B8, 0019,2034,4DC0,0800,0000,11B9  ;2159  q_q.and.k[.ff00] ; ...
U 11B9, 0019,2032,1D80,09F8,0000,0001  ;2160  rc[0f]_q.or.sc,return1 ; Set new argument count and return
            ;2161
```

```
            ;2162  .PAGE
            ;2163  .TOC "   Increment Subtest Number"
            ;2164 ;++
            ;2165 ; FUNCTIONAL DESCRIPTION:
            ;2166 ;
            ;2167 ; This routine is used to increment the subtest number
            ;2168 ; in RC[0F]<15:8>.
            ;2169 ;
            ;2170 ; CALLING CONSTRAINT:
            ;2171 ;
            ;2172 ; =0
            ;2173 ;
            ;2174 ; SIDE EFFECTS:
            ;2175 ;
            ;2176 ; D register is destroyed
            ;2177 ;--
            ;2178 subtst:
U 11BA, 0810,0038,4D80,0978,0000,11BB  ;2179  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2180  rc[0f]_d+k[.ff00], ; Increment and return
U 11BB, 0019,4016,4D80,09F8,0000,0001  ;2181      word,return1 ; (Subtest number is negative)
            ;2182
```

```
            ;2183  .PAGE
            ;2184  .TOC " Diagnostic Specific Subroutines"
            ;2185  .TOC "   Select Controller"
            ;2186 ;**
            ;2187 ; FUNCTIONAL DESCRIPTION:
            ;2188 ;
            ;2189 ; This routine is used to put the memory system into the
            ;2190 ; specified configuration with respect to controller select.
            ;2191 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2192 ; a RETURN2 is made.
            ;2193 ;
            ;2194 ; CALLING CONSTRAINT:
            ;2195 ;
            ;2196 ; =00
            ;2197 ;
            ;2198 ; INPUTS:
            ;2199 ;
            ;2200 ; d - Contains value to write to bits<2:0> of Register A
            ;2201 ; rc[0e] - Address of Register A
            ;2202 ;
            ;2203 ; SIDE EFFECTS:
            ;2204 ;
            ;2205 ; sc,d,va are destroyed
            ;2206 ;--
            ;2207 SELECT.CNTRLR:
U 11BC, 0010,003B,0180,0970,0284,71BD  ;2208  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11BD, 0801,001C,0180,0800,0000,11BE  ;2209  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11BE, 0000,003C,0180,A800,0000,11BF  ;2210  cache.p_d[long]  ; Write the configuration Register
U 11BF, 081B,003B,5D80,0800,0000,11C0  ;2211  d_k[.7]   ; Get Misconfiguration bits
U 11C0, 000D,003C,61F8,0800,0084,71C1  ;2212  sc_k[.F],q_0  ; ...
U 11C1, 0D00,003C,0180,0800,0000,11C2  ;2213  d_dal.sc  ; ... D = x38000
U 11C2, 0000,003C,01E0,0800,0000,11C3  ;2214  q_d   ; Save mask
U 11C3, 0000,003C,0180,C800,0000,11C4  ;2215  d[long]_cache.p  ; Read CNFG A Reg and
            ;2216  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11C4, 001D,2034,0180,0800,0010,11C5  ;2217   clk.ubcc ; ...
U 11C5, 0000,013C,0180,0800,0000,1058  ;2218  z?   ; Branch if no
U 1058, 0000,003E,0180,0800,0000,0001  ;2219 =0 RETURN1   ; Bad configuration
U 1059, 0000,003E,0180,0800,0000,0002  ;2220  RETURN2   ; Configuration ok
            ;2221
```

```
;2222  .PAGE
;2223  .TOC "    64K OR 256K CHIPS"
;2224 ;*************************************************************
;2225 ;**
;2226 ;
;2227 ; Functional Description:
;2228 ; ----------------------
;2229 ; This subroutine is used to find the type of chips
;2230 ; on the MS780-E/H arrays.  The RETURN modes are as
;2231 ; follows:
;2232 ;
;2233 ; RETURN1 = Error. Mixed 64K and 256K arrays
;2234 ;
;2235 ; RETURN2 = 64K chips on the array
;2236 ;
;2237 ; RETURN3 = 256K chips on the array
;2238 ;
;2239 ; Calling Constraint: =00
;2240 ; ------------------
;2241 ;
;2242 ;
;2243 ; Inputs:
;2244 ; ------
;2245 ;
;2246 ; RC[0E] must hold the I/O base address of the MS780-E/H
;2247 ; currently under test
;2248 ;
;2249 ; Outputs:
;2250 ; -------
;2251 ;
;2252 ; NO specific outputs. (See RETURN modes above)
;2253 ;
;2254 ;
;2255 ; Side Effects:
;2256 ; ------------
;2257 ;
;2258 ; The contents of the following registers will be lost:
;2259 ;
;2260 ; D
;2261 ;
;2262 ;--
;2263 ;*************************************************************
;2264 64K.OR.256K:
U 11C6, 0010,0038,0180,0970,0200,11C7  ;2265  VA_RC[0E]    ; Point to CNFG Register A
U 11C7, 0000,003C,0180,C800,0000,11C8  ;2266  D[LONG]_CACHE.P    ; Read the register
U 11C8, 0200,003C,0180,0800,0000,11C9  ;2267  D_D.RIGHT2    ; Shift received contents two bits
        ;2268        ; ...to the right
U 11C9, 0600,003C,0180,0800,0000,11CA  ;2269  D_D.RIGHT    ; Shift D one more time to bring bits
        ;2270        ; ...in position
U 11CA, 0000,193C,0180,0800,0000,100C  ;2271  D1-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2272 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2273  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2274  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2275  RETURN1     ; Error: missing arrays
        ;2276
```

```
;2277   .PAGE
;2278   .TOC "   START TEST"
;2279 ;*************************************************************
;2280 ;**
;2281 ;
;2282 ; Functional Description:
;2283 ; ---------------------
;2284 ;
;2285 ;         This subroutine will be called by all tests that  check  the
;2286 ;    controllers,  or  those  tests  that  have  to switch between the
;2287 ;    controllers when executing.   Its  main  functions  are:   select
;2288 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2289 ;    test that called it, call out an error  if  neither  one  of  the
;2290 ;    controllers can be configured properly, if all the controllers on
;2291 ;    a MS780-E/H were configured and tested, it should select the next
;2292 ;    MS780-E/H on the SBI and repeat the above sequence.
;2293 ;    A test making use of this  subroutine  should  be  configured  as
;2294 ;    follows:
;2295 ;
;2296 ;
;2297 ;       Begin TEST.XX
;2298 ;        :
;2299 ;        :   Call NEWTST
;2300 ;        :   Call FIND.MS780E
;2301 ;        :   IF No MS780-E's on the SBI
;2302 ;        :     THEN
;2303 ;        :         Skip to End of TEST.XX
;2304 ;        :     ELSE
;2305 ;        :
;2306 ;        : RESTART.TEST.XX:
;2307 ;        :
;2308 ;        :   Call START TEST
;2309 ;        :   IF Return1 from START.TEST
;2310 ;        :     THEN
;2311 ;        :         Skip to End of TEST.XX
;2312 ;        :     ELSE.
;2313 ;        :
;2314 ;        ~
;2315 ;
;2316 ;        :
;2317 ;        :   Body of TEST.XX
;2318 ;        :
;2319 ;        ~
;2320 ;
;2321 ;        :   Jump RESTART.TEST.XX
;2322 ;        :
;2323 ;       End TEST.XX
;2324 ;
;2325 ;
;2326 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2327 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2328 ;    should be passed when the subroutine is called a second, third or
;2329 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2330 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2331 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2332 ;    Register 39 can be interpreted as follows:
;2333 ;
;2334 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2335 ;               called the first time.  When this code is encountered
;2336 ;               this subroutine will try to select the lower controller.
;2337 ;               If the lower controller is misconfigured, the pointer in
;2338 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2339 ;               is re-entered.
;2340 ;               If, however there is no misconfiguration, the value in
;2341 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2342 ;               calling test.
;2343 ;
;2344 ;     b01   -   This value signifies that an attempt at configuring the
;2345 ;               lower controller failed, and the subroutine will attempt
;2346 ;               to select the upper controller.
;2347 ;               If the upper controller is also misconfigured execution
;2348 ;               stops with a call to ERROR1.
;2349 ;               If there is no misconfiguration on this controller, then
;2350 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2351 ;               made to the calling test.
;2352 ;
;2353 ;     b10   -   This value signifies that the lower controller was selected
;2354 ;               previously and the test was executed once.  If the
;2355 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2356 ;               39 is loaded with b11 and an attempt is made to select the
;2357 ;               upper controller.
;2358 ;               If there is a misconfiguration on this controller, this
;2359 ;               subroutine is just re-entered.
;2360 ;               Otherwise, a Return2 will be made to the calling test.
;2361 ;
;2362 ;     b11   -   This code identifies the cases in which the test has
;2363 ;               been executed at least once (i.e., at least one of the
;2364 ;               controllers on the MS780-E unit under test could be
;2365 ;               configured properly).  When this code is detected the
;2366 ;               subroutine clears ID Reg 39 and makes a call to
;2367 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2368 ;               tested then the subroutine executes a Return1.
;2369 ;               Otherwise the subroutine is re-entered.
;2370 ;
;2371 ;
;2372 ; Calling Constraint: =00
;2373 ; ------------------
;2374 ;
;2375 ;
;2376 ; Inputs:
;2377 ; ------
;2378 ;
;2379 ; ID Register 39 must be cleared by NEWTST
;2380 ;
;2381 ;
;2382 ; Outputs:
;2383 ; -------
;2384 ;
;2385 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2386 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2387 ;         to be tested
            ;2388 ;
            ;2389 ; Side Effects:
            ;2390 ; ------------
            ;2391 ;
            ;2392 ; Contents of the following regiters will be destroyed:
            ;2393 ;
            ;2394 ; D, Q
            ;2395 ;
            ;2396 ;--
            ;2397 ;*************************************************************
            ;2398 =0
            ;2399 START.TEST:
U 105C, 0000,003D,0180,0800,0000,11D3  ;2400  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 105D, 0000,003C,0180,0800,0000,106C  ;2401  NOP    ; ...tests that have more than one
            ;2402 =0      ; ...subtest.)
U 106C, 0000,003D,0180,0800,0000,11B2  ;2403  ERLOOP    ; Set up the error loop for the
            ;2404       ; ...eventuality that the MS780-E/H
            ;2405       ; ...currently under test has no
            ;2406       ; ...Controllers
            ;2407 RE.ENTRY:    ; Re entry point for this routine
U 106D, 0000,003C,E5F0,2C00,0000,11CB  ;2408  Q_ID[39]    ; Get the code
U 11CB, 0C00,003C,0180,0800,0000,11CC  ;2409  D_Q    ; Put it in the D Register
U 11CC, 0000,193C,0180,0800,0000,101C  ;2410  D1-0?    ; Branch on the code
U 101C, 0000,003C,0180,0800,0000,1008  ;2411 =1100 J/STRT.CASE00   ; Code is 00
U 101D, 0000,003C,0180,0800,0000,1010  ;2412  J/STRT.CASE01   ; Code is 01
U 101E, 0000,003C,0180,0800,0000,11D1  ;2413  J/STRT.CASE10   ; Code is 10
U 101F, 0000,003C,0180,0800,0000,1017  ;2414  J/STRT.CASE11   ; Code is 11
            ;2415 ;
            ;2416 ;
            ;2417 ; At this point the code in ID[39] was 00
            ;2418 ;
            ;2419 ;
            ;2420 =00
            ;2421 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1020  ;2422  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2423  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,11CD  ;2424   D_K[.1]    ; Set up the code for re entry to this
            ;2425       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2426  D_K[.2]    ; Set up the code for a next call to
            ;2427       ; ...START.TEST indicating that the
            ;2428       ; ...Lower Controller was configured
            ;2429       ; ...( 10 )
            ;2430  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2431   RETURN2    ; Let the test know that the Lower
            ;2432       ; ...has been configured
            ;2433 STRT.LOW.MIS:
            ;2434  ID[39]_D,    ; Save code in ID[39] signifing that
U 11CD, 0000,003C,E580,3C00,0000,106D  ;2435   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2436       ; ...and re enter this subroutine
            ;2437
            ;2438 ;
            ;2439 ;
            ;2440 ; At this point the code is 01
            ;2441 ;
```

```
            ;2442 ;
            ;2443 =00
            ;2444 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1024  ;2445  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2446  J/STRT.UPR.MIS.    ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,11CE  ;2447   D_0    ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2448  D_K[.3]    ; Upper controller was configured
            ;2449       ; ...thus the code must be changed
            ;2450       ; ...accordingly ( 11 )
            ;2451  ID[39]_D,    ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2452   RETURN2    ; Let the test know that the Upper
            ;2453       ; ...Controller has been configured
            ;2454 STRT.UPR.MIS:
U 11CE, 0000,003C,E580,3C00,0000,11CF  ;2455  ID[39]_D    ; Save the Code ( 00 )
U 11CF, 0018,0038,0D80,09E8,0000,11D0  ;2456  RC[0D]_K[.3]    ; Set up a code for the Fail Chain
U 11D0, 0000,003D,0980,0800,0084,704C  ;2457  ERROR1[.2]    ; Flag the error.
            ;2458 ;
            ;2459 ;
            ;2460 ; At this point the code is 10
            ;2461 ;
            ;2462 ;
            ;2463 STRT.CASE10:
U 11D1, 0818,0038,0D80,0800,0000,1014  ;2464  D_K[.3]    ; Set the code to 11
            ;2465
            ;2466
            ;2467 =00 ID[39]_D,    ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1024  ;2468   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,106D  ;2469  J/RE.ENTRY    ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2470  RETURN2     ; Upper Controller configured
            ;2471       ; ...let the test know it
            ;2472 ;
            ;2473 ;
            ;2474 ; At this point the code is 11
            ;2475 ;
            ;2476 ;
            ;2477 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1018  ;2478  D_0     ; Clear the code
            ;2479 =00 ID[39]_D,    ; Save the code
U 1018, 0000,003D,E580,3C00,0000,1222  ;2480   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2481       ; ...on the SBI
U 1019, 0000,003C,0180,0800,0000,106D  ;2482  J/RE.ENTRY    ; Found another MS780-E/H.   Go and
            ;2483       ; ...attempt to configure the cntrllrs
U 101A, 0000,003E,0180,0800,0000,0001  ;2484  RETURN1     ; No more MS780-E/Hs found
U 101B, 0000,003C,0180,0800,0000,1020  ;2485  NOP
            ;2486
```

```
            ;2487  .PAGE
            ;2488  .TOC "   SELECT LOWER CONTROLLER"
            ;2489 ;*************************************************************
            ;2490 ;++
            ;2491 ;
            ;2492 ; Functional Description:
            ;2493 ; ----------------------
            ;2494 ;
            ;2495 ; This subroutine can be called to select the lower
            ;2496 ; Controller on a MS780-E/H.
            ;2497 ; If the Lower controller is misconfigured then a
            ;2498 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2499 ;
            ;2500 ; Calling Constraint: =00
            ;2501 ; ------------------
            ;2502 ;
            ;2503 ;
            ;2504 ; Inputs:
            ;2505 ; ------
            ;2506 ;
            ;2507 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2508 ;
            ;2509 ; Outputs:
            ;2510 ; -------
            ;2511 ;
            ;2512 ; RC[0D] will have the address of CNFG Register C
            ;2513 ;        ( 2000x008 )
            ;2514 ;
            ;2515 ; Side Effects:
            ;2516 ; ------------
            ;2517 ;
            ;2518 ; Contents of the following registers will lost:
            ;2519 ;
            ;2520 ; D
            ;2521 ;
            ;2522 ; The Lower controller on the MS780-E/H will be configured
            ;2523 ; when the subroutine is exitted with a RETURN2
            ;2524 ;--
            ;2525 ;*************************************************************
            ;2526 =00
            ;2527 SELECT.LOWER:
            ;2528  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1020, 0818,0039,1980,0800,0000,11BC  ;2529   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000,003E,0180,0800,0000,0001  ;2530  RETURN1     ; Lower Controller Misconfigured
U 1022, 0810,0038,0180,0970,0000,1023  ;2531  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2532  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1023, 0019,0016,0180,09E8,0000,0002  ;2533   RETURN2    ; Let caller know that the controller
            ;2534       ; ...was configured properly
```

```
;2535  .PAGE
;2536  .TOC "   SELECT UPPER CONTROLLER"
;2537 ;*************************************************************
;2538 ;++
;2539 ;
;2540 ; Functional Description:
;2541 ; ----------------------
;2542 ;
;2543 ; This subroutine can be called to select the upper
;2544 ; Controller on a MS780-E/H.
;2545 ; If the Upper controller is misconfigured then a
;2546 ; RETURN1 will be made.  Otherwise a RETURN2
;2547 ;
;2548 ; Calling Constraint: =00
;2549 ; ------------------
;2550 ;
;2551 ;
;2552 ; Inputs:
;2553 ; ------
;2554 ;
;2555 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2556 ;
;2557 ; Outputs:
;2558 ; -------
;2559 ;
;2560 ; RC[0D] will have the address of CNFG Register D
;2561 ;        ( 2000x010 )
;2562 ;
;2563 ; Side Effects:
;2564 ; ------------
;2565 ;
;2566 ; Contents of the following registers will lost:
;2567 ;
;2568 ; D
;2569 ;
;2570 ; The Upper controller on the MS780-E/H will be configured
;2571 ; when the subroutine is exitted with a RETURN2
;2572 ;--
;2573 ;*************************************************************
;2574 =00
;2575 SELECT.UPPER:
;2576  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,0980,0800,0000,11BC  ;2577   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2578  RETURN1     ; Upper Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2579  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2580  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,8580,09E8,0000,0002  ;2581   RETURN2    ; Let caller know that the controller
;2582       ; ...was configured properly
;2583
```

```
            ;2584  .PAGE
            ;2585  .TOC "    SBI Unjam Routine"
            ;2586 ;**
            ;2587 ; FUNCTIONAL DESCRIPTION:
            ;2588 ;
            ;2589 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2590 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2591 ; and finally HOLD for the last 16 cycles.
            ;2592 ;
            ;2593 ; CALLING CONSTRAINT:
            ;2594 ;
            ;2595 ; =0
            ;2596 ;
            ;2597 ; SIDE EFFECTS:
            ;2598 ;
            ;2599 ; D Reg destroyed
            ;2600 ;--
            ;2601    ;
            ;2602    ; assert hold for 16 cycles
            ;2603    ;
            ;2604 CLRSBI:
            ;2605 SBI.UNJAM:
            ;2606  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11D2, 0818,0038,6580,8000,0010,10AC  ;2607      clk.ubcc  ; set micro cc's for next cycle
            ;2608 =0
            ;2609 SBI.UNJAM.1:
            ;2610  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2611       clk.ubcc,z?, ; ...
U 10AC, 0819,0100,0580,8000,0010,10AC  ;2612      j/sbi.unjam.1 ; test for completion
            ;2613    ;
            ;2614    ; assert hold and unjam for 16 cycles
            ;2615    ;
            ;2616  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10AD, 0818,0038,6580,8800,0010,10BC  ;2617      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2618 =0
            ;2619 SBI.UNJAM.2:
            ;2620  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2621       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10BC, 0819,0100,0580,8800,0010,10BC  ;2622      z?,j/sbi.unjam.2 ; ...
            ;2623    ;
            ;2624    ; assert hold for 16 cycles
            ;2625    ;
            ;2626  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10BD, 0818,0038,6580,8000,0010,10C4  ;2627      clk.ubcc  ; and set cc's for next cycle
            ;2628 =0
            ;2629 SBI.UNJAM.3:
            ;2630  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2631       clk.ubcc,z?, ; ...
U 10C4, 0819,0100,0580,8000,0010,10C4  ;2632      j/sbi.unjam.3 ; ...
U 10C5, 0000,003E,0180,0800,0000,0001  ;2633  return1   ; exit
            ;2634
```

```
          ;2635  .PAGE
          ;2636  .TOC "   RESET SUBTEST NUMBER"
          ;2637 ;**
          ;2638 ; FUNCTIONAL DESCRIPTION:
          ;2639 ;
          ;2640 ; Since some  of the tests will have to be restarted when two
          ;2641 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
          ;2642 ; the subtest counter to zero ( only for thoes tests that have
          ;2643 ; more than one subtest). This subroutine may also be necessary
          ;2644 ; when a test is being loop on.
          ;2645 ;
          ;2646 ; CALLING CONSTRAINT:
          ;2647 ;
          ;2648 ; =0
          ;2649 ; SIDE EFFECTS:
          ;2650 ;
          ;2651 ; D is destroyed
          ;2652 ;
          ;2653 ;--
          ;2654 RESET.SUBTST:
          ;2655  RC[0F]_0,   ; Reset subtest number
U 11D3, 0003,003E,0180,09F8,0000,0001  ;2656   RETURN1   ; ...and return
```

```
            ;2657  .PAGE
            ;2658  .TOC "   Initialize the SBI"
            ;2659 ;++
            ;2660 ; FUNCTIONAL DESCRIPTION:
            ;2661 ;
            ;2662 ; This routine performs the following functions:
            ;2663 ;
            ;2664 ;  Executes an SBI Unjam
            ;2665 ;  Clears the SBI Fault Latch
            ;2666 ;  Clears the SBI Error Register
            ;2667 ;
            ;2668 ; CALLING CONSTRAINT:
            ;2669 ;
            ;2670 ; =0
            ;2671 ;
            ;2672 ; SIDE EFFECTS:
            ;2673 ;
            ;2674 ; D & Q Register destroyed
            ;2675 ;--
            ;2676 =0
            ;2677 SBI.INIT:
U 10CC, 0000.003D.0180.0800.0000.11D2  ;2678  call[sbi.unjam]  ; Unjam the SBI
U 10CD, 0818.0038.C180.0800.0000.11D4  ;2679  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11D4, 0801.0028.6580.3C00.0000.11D5  ;2680  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11D5, 0F00.003C.6D80.3C00.0000.11D6  ;2681  id[fault]_d,d_0
U 11D6, 0000.003C.6D80.3C00.0000.11D7  ;2682  id[fault]_d
U 11D7, 0000.003E.6580.3C00.0000.0001  ;2683  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2684
```

```
            ;2685  .PAGE
            ;2686  .TOC "   Enable Cache"
            ;2687 ;**
            ;2688 ; FUNCTIONAL DESCRIPTION:
            ;2689 ;
            ;2690 ; This routine enables cache group 0 by clearing the force
            ;2691 ; miss bit in the maintenance register.
            ;2692 ;
            ;2693 ; CALLING CONSTRAINT:
            ;2694 ;
            ;2695 ; =0
            ;2696 ;
            ;2697 ; SIDE EFFECTS:
            ;2698 ;
            ;2699 ; D, Q AND SC Registers destroyed
            ;2700 ;--
            ;2701 ENABLE.CACHE:
U 11D8, 0000,003C,75F0,2C00,0000,11D9  ;2702  q_id[maint]  ; Get current maintenance register
U 11D9, 0000,003C,6580,0800,0084,71DA  ;2703  sc_k[.10]  ; Setup to clear bit 16
U 11DA, 0801,2034,0180,0800,0000,11DB  ;2704  d_q.and.mask  ; Clear bit 16
U 11DB, 0000,003E,7580,3C00,0000,0001  ;2705  id[maint]_d,return1 ; Rewrite register and return
            ;2706
```

```
            ;2707  .PAGE
            ;2708  .TOC "   Disable Cache Routine"
            ;2709 ;++
            ;2710 ; FUNCTIONAL DESCRIPTION:
            ;2711 ;
            ;2712 ; This routine disables cache hits by setting bits <16:15>
            ;2713 ; in the maintenance register.
            ;2714 ;
            ;2715 ; CALLING SEQUENCE:
            ;2716 ;
            ;2717 ; =0
            ;2718 ;
            ;2719 ; SIDE EFFECTS:
            ;2720 ;
            ;2721 ; D & Q Registers destroyed
            ;2722 ;--
            ;2723 DISABLE.CACHE:
U 11DC, 0818,0038,4580,0800,0000,11DD  ;2724  d_k[.8000]  ; ... and seed constant
U 11DD, 0500,003C,0180,0800,0000,11DE  ;2725  d_d.left  ; Generate final constant
U 11DE, 0819,0030,4580,0800,0000,11DF  ;2726  d_d.or.k[.8000]  ; ... = 00018000
U 11DF, 0000,003E,7580,3C00,0000,0001  ;2727  id[maint]_d,return1 ; Disable cache and return
            ;2728
```

```
            ;2729  .PAGE
            ;2730  .TOC "   Check for Interrupt Routine"
            ;2731 ;++
            ;2732 ; FUNCTIONAL DESCRIPTION:
            ;2733 ;
            ;2734 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2735 ; If an interrupt is active, the following registers are setup:
            ;2736 ;  RC[8] - Gets the VECTOR register
            ;2737 ;  RC[7] - Gets the SBI ERROR register
            ;2738 ;  RC[6] - Gets the FAULT register
            ;2739 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2740 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2741 ;
            ;2742 ; CALLING CONSTRAINT:
            ;2743 ;
            ;2744 ; =00
            ;2745 ;
            ;2746 ; SIDE EFFECTS:
            ;2747 ;
            ;2748 ; D & Q are destroyed
            ;2749 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2750 ;--
            ;2751 CHECK_INTERRUPT:
            ;2752   ;
            ;2753   ; First, enable the fault and RDS/CRD interrupts
            ;2754   ;
U 11E0, 0818,0038,4580,0800,0000,11E1  ;2755  d_k[.8000]  ; Get data to set interrupt enable
            ;2756  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 11E1, 0500,003C,6580,3C00,0000,11E2  ;2757      d_d.left
U 11E2, 0100,003C,0180,0800,0000,11E3  ;2758  d_d.left2  ; D = 40000
U 11E3, 0000,003C,6D80,3C00,0000,11E4  ;2759  id[fault]_d  ; Set fault interrupt enable
            ;2760  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 11E4, 0F03,003C,0180,09A8,4000,11E5  ;2761      rc[5]_0  ; and clear interrupt occurred flag
U 11E5, 0000,003C,3D80,3C00,0000,11E6  ;2762  id[psl]_d  ; Clear the PSL
U 11E6, 0000,003C,6580,3C00,0000,11E7  ;2763  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 11E7, 0000,003C,6D80,3C00,0000,11E8  ;2764  id[fault]_d  ; Clear FAULT Interrupt Enable
U 11E8, 0000,0E3C,0180,0800,0000,1004  ;2765  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2766 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2767  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2768  return1   ; No interrupt
            ;2769 =111
            ;2770 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,11E9  ;2771  q_id[vector]  ; Get ID Vector Register
U 11E9, 0001,203C,65F0,2DC0,0000,11EA  ;2772  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 11EA, 0001,203C,6DF0,2D88,0000,11EB  ;2773  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 11EB, 0001,203C,0180,[illegible]B0,0000,11EC  ;2774  rc[6]_q   ; Save falut reg
U 11EC, 0018,003A,0580,09A8,0000,0002  ;2775  rc[5]_k[.1],return2 ; Set error flag and return
            ;2776
```

```
            ;2777  .PAGE
            ;2778  .TOC "   Set Trap Mask"
            ;2779 ;++
            ;2780 ; FUNCTIONAL DESCRIPTION:
            ;2781 ;
            ;2782 ; This routine is used to set a bit in the Micro Trap Mask
            ;2783 ; R[0F].
            ;2784 ;
            ;2785 ; CALLING CONSTRAINT:
            ;2786 ;
            ;2787 ; =0
            ;2788 ;
            ;2789 ; INPUTS:
            ;2790 ;
            ;2791 ; SC - Contains bit number to set.
            ;2792 ;
            ;2793 ; OUTPUTS:
            ;2794 ;
            ;2795 ; rc[0E] - Contains the current trap mask
            ;2796 ;
            ;2797 ; SIDE EFFECTS:
            ;2798 ;
            ;2799 ; d regsiter is destroyed
            ;2800 ;--
            ;2801 SET_TRAP_MASK:
U 11ED, 0800,003C,0180,0A78,0000,11EE  ;2802  d_r[0f]
U 11EE, 0801,001C,0180,0AF8,0000,11EF  ;2803  r[0f]_d.ornot.mask,d_alu ; Set mask
U 11EF, 0001,003E,0180,0AF0,0000,0001  ;2804  r[0E]_d,return1    ; Save mask and return
            ;2805
```

```
            ;2806  .PAGE
            ;2807  .TOC "   FIND MS780-E MEMORY"
            ;2808 ;++
            ;2809 ; FUNCTIONAL DESCRIPTION:
            ;2810 ;
            ;2811 ;          The diagnostic is designed to handle up to 4 (four)
            ;2812 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2813 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2814 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2815 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2816 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2817 ;     first MS780-E found will have its starting address set to 0
            ;2818 ;     (zero).
            ;2819 ;     All the other MS780-E/H's found will have their starting address
            ;2820 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2821 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2822 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2823 ;     current MS780-E unit under test.
            ;2824 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2825 ;     set to maximum.
            ;2826 ;
            ;2827 ; CALLING CONSTRAINT:
            ;2828 ;
            ;2829 ; =00
            ;2830 ;
            ;2831 ; OUTPUTS:
            ;2832 ;
            ;2833 ; rc[0e] - Contains address of Configuration Register
            ;2834 ; r[0]  - Contains TR level
            ;2835 ;
            ;2836 ; SIDE EFFECTS:
            ;2837 ;
            ;2838 ; D register is destroyed.
            ;2839 ;--
            ;2840 =0
            ;2841 FIND.MS780E:
U 10D0, 0000,003D,0180,0800,0000,10CC  ;2842  call[sbi.init]  ; Make sure SBI is enabled
U 10D1, 0003,003C,0180,0988,0000,11F0  ;2843  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11F0, 0818,0038,1980,0800,0000,11F1  ;2844  d_k[zero]  ; Clear MS780-E/H counter
U 11F1, 0000,003C,F980,3C00,0000,11F2  ;2845  id[3e]_d  ; ...
U 11F2, 0018,0038,0580,0A80,0000,11F3  ;2846  r[0]_k[.1]  ; Init TR counter
U 11F3, 0F03,003C,0180,09F0,0000,10D2  ;2847  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2848     ; ...CNFG A Reg
            ;2849 =0
            ;2850 FIND.MEM.LOOP:
U 10D2, 0800,003D,0180,0A00,0000,1219  ;2851  d_r[0],call[generate.io]; Generate address of TR level
U 10D3, 0001,003C,0180,09F0,0200,10D4  ;2852  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10D4, 0000,003D,8980,0800,0084,71ED  ;2853 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2854  d[long]_cache.p, ; Read the specified tr level
U 10D5, 0000,003C,0180,C970,0000,11F4  ;2855   ic_rc[0e]  ; ...
U 11F4, 0000,003C,0180,0A78,0010,11F5  ;2856  alu_r[0f],clk.ubcc ; Check for no response
U 11F5, 0001,013C,0180,0880,0082,10D6  ;2857  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10D6, 0000,003C,0180,0800,0000,11F6  ;2858 =0 j/check.adpt.code ; Check for a memory
U 10D7, 0000,003C,0180,0800,0000,1215  ;2859  j/find.mem.lp1  ; Try next tr
            ;2860 CHECK.ADPT.CODE:
```

```
U 11F6, 0000,003C,1D80,0800,1404,71F7  ;2861  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11F7, 0000,163C,0180,0800,0000,1028  ;2862  state7-4?  ; Branch on the adapter type
U 1028, 0000,003C,8980,0800,0084,71F8  ;2863 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1029, 0000,003C,8980,0800,0084,71F9  ;2864  j/ms780.c,sc_k[.d] ; MS780-C
U 102A, 0000,003C,0180,0800,0000,1215  ;2865  j/find.mem.lp1  ; Not a memory
U 102B, 0000,003C,0180,0800,0000,1215  ;2866  j/find.mem.lp1  ; Not a memory
U 102C, 0000,003C,6580,0800,0084,71FE  ;2867  j/ma780.max,sc_k[.10] ; MA780
U 102D, 0000,003C,0180,0800,0000,1215  ;2868  j/find.mem.lp1  ; Not a memory
U 102E, 0010,0038,0180,09F0,0000,1202  ;2869  rc[0e]_lc,j/found.ms780e ; MS780-E
U 102F, 0010,0038,0180,09F0,0000,1202  ;2870  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2871   ;
            ;2872   ; Found an MS780-A or -C. Set the starting address
            ;2873   ; to maximum.
            ;2874   ;
U 11F8, 0818,0038,5D80,0800,0000,11FA  ;2875 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11F9, 0818,0038,0580,0800,0000,11FA  ;2876 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2877 MS780.COMMON:
U 11FA, 0819,0030,7180,0800,0000,11FB  ;2878  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11FB, 0000,003C,01F8,0803,0080,D1FC  ;2879  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11FC, 0D00,003C,0180,0800,0000,11FD  ;2880  d_del.sc  ; Put data in bits<29:14>
            ;2881  cache.p_d[long], ; Set the starting address to maximum
U 11FD, 0000,003C,0180,A800,0000,1215  ;2882       j/find.mem.lp1 ; and continue TR scan
            ;2883   ;
            ;2884   ; Found an MA780. Set the starting address to maximum
            ;2885   ;
            ;2886 MA780.MAX:
U 11FE, 0818,0038,39F8,0803,0000,11FF  ;2887  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11FF, 0D00,003C,0180,0803,0000,1200  ;2888  d_del.sc,va_va+4 ; Put in proper position
U 1200, 0000,003C,0180,0803,0000,1201  ;2889  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2890  cache.p_d[long], ; Set starting address to maximum
U 1201, 0000,003C,0180,A800,0000,1215  ;2891       j/find.mem.lp1
            ;2892 ;
            ;2893 ; Found an MS780E
            ;2894 ;
            ;2895 FOUND.MS780E:
U 1202, 0000,003C,F9F0,2C00,0000,1203  ;2896  q_id[3e]  ; Set up to see how many MS780-E's
            ;2897  alu_q.xor.k[.3], ; Are there Maximum units?
U 1203, 0019,2020,0D80,0800,0010,1204  ;2898   clk.ubcc ; Check condition codes
U 1204, 0000,013C,0180,0800,0000,10D8  ;2899  z?   ; Are we at maximum units for system
U 10D8, 0000,003C,0180,0800,0000,1205  ;2900 =0 J/ms780e.tst  ; No go find how many units ther are
U 10D9, 0818,0038,C180,0800,0000,10DA  ;2901  d_k[.ffff]  ; Yes set address to maximum
U 10DA, 0000,003D,0180,0800,0000,121D  ;2902 =0 call[set.start.addr] ; .....
U 10DB, 0000,003C,0180,0800,0000,1215  ;2903  j/find.mem.lp1  ; Go check to see if we are done
            ;2904 ;
            ;2905 MS780E.TST:
            ;2906  alu_rc[01],  ; Check "Found MS780E Flag"
U 1205, 0010,0038,0180,0908,0010,1206  ;2907   clk.ubcc ; Check condition codes
U 1206, 0810,0138,0180,0970,0000,10DC  ;2908  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2909 =0 j/not.first.ms780e, ; No
U 10DC, 0818,0038,C180,0800,0000,10E0  ;2910   d_k[.ffff] ; Set starting address
U 10DD, 0001,003C,0180,0988,0000,1207  ;2911  rc[01]_d  ; Yes put base address in rc[01]
U 1207, 0818,0038,7980,0800,0000,1208  ;2912  d_k[.30]  ; Set up SC to point to first unit
U 1208, 0019,0014,F580,0800,0082,1209  ;2913  alu_d+k[.a],sc_alu ; ...
U 1209, 0810,0038,0180,0908,0000,120A  ;2914  d_rc[01]  ; Put base address of unit in D
U 120A, 0000,003C,0180,3400,0000,120B  ;2915  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 120B, 0F00,003C,0180,0800,0000,10DE  ;2916  d_0   ; Prepare to set starting address to Zero
U 10DE, 0000,003D,0180,0800,0000,121D  ;2917 =0 call[set.start.addr] ; Go do it
          ;2918  j/find.mem.lp1,  ; Check to see if we are done
U 10DF, 0003,003C,0180,09E8,0000,1215  ;2919   rc[0d]_0 ; ....
          ;2920 =0
          ;2921 NOT.FIRST.MS780E:
U 10E0, 0000,003D,0180,0800,0000,121D  ;2922  call[set.start.addr] ; Go set starting address to maximum
U 10E1, 0000,003C,F9F0,2C00,0000,120C  ;2923  q_id[3e]  ; Get the number of MS780-Es found
U 120C, 0819,2014,0580,0800,0000,120D  ;2924  d_q+k[.1]  ; Increment this counter
U 120D, 0000,003C,F980,3C00,0000,120E  ;2925  id[3e]_d  ; Save the counter
U 120E, 0018,0038,11C0,0800,0000,120F  ;2926  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2927     ; ...were the I/O base addresses will be stored
U 120F, 081D,2000,7980,0800,0000,1210  ;2928  d_q-d,k[.30]  ; ... adjust pointer
U 1210, 0819,0014,7980,0800,0000,1211  ;2929  d_d+k[.30]  ; Point the SC to the ID register
U 1211, 0000,003C,F580,0800,0000,1212  ;2930  k[.a]   ; ...to receive I/O base address
U 1212, 0019,0014,F580,0800,0082,1213  ;2931  alu_d+k[.a],sc_alu ; ...
U 1213, 0810,0038,0180,0970,0000,1214  ;2932  d_rc[0e]  ; Get base address to d
U 1214, 0000,003C,0180,3400,0000,1215  ;2933  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2934   ;
          ;2935   ; Try next tr
          ;2936   ;
          ;2937 FIND.MEM.LP1:
U 1215, 0818,0014,0580,0A80,0000,1216  ;2938  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2939  alu_d.and.k[.f],
U 1216, 0019,0034,6180,0800,001D,1217  ;2940      clk.ubcc  ; Check if all done
U 1217, 0000,013C,0180,0800,0000,10E2  ;2941  z?   ; Branch if yes
U 10E2, 0000,003C,0180,0800,0000,10D2  ;2942 =0 j/find.mem.loop  ; Try next TR
U 10E3, 0810,0038,0180,0908,001D,1218  ;2943  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1218, 0000,013C,0180,0800,0000,10E4  ;2944  z?   ; Check condition codes
U 10E4, 0001,003E,0180,09F0,000D,0002  ;2945 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10E5, 0000,003E,0180,0800,0000,0001  ;2946  return1   ; No MS780E found
          ;2947
```

```
            ;2948  .PAGE
            ;2949  .TOC "   Generate IO Address"
            ;2950 ;++
            ;2951 ; FUNCTIONAL DESCRIPTION:
            ;2952 ;
            ;2953 ; This routine is used to generate an SBI Configuration Register
            ;2954 ; address from a TR level.
            ;2955 ;
            ;2956 ; CALLING CONSTRAINT:
            ;2957 ;
            ;2958 ; =0
            ;2959 ;
            ;2960 ; INPUTS:
            ;2961 ;
            ;2962 ; D - Contains TR level
            ;2963 ;
            ;2964 ; OUTPUTS:
            ;2965 ;
            ;2966 ; D - Contains SBI IO address
            ;2967 ;--
            ;2968 GENERATE.IO:
U 1219, 0000,003C,89F8,0800,0084,721A  ;2969  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 121A, 0D00,003C,8D80,0800,0084,721B  ;2970  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 121B, 0000,003C,0980,0800,0084,B21C  ;2971  sc_sc-k[.2]  ; insert bit 29
U 121C, 0801,001E,0180,0800,0000,0001  ;2972  d_d.ornot.mask,return1 ; and return
            ;2973
```

```
            ;2974  .PAGE
            ;2975  .TOC "   Set Starting Address"
            ;2976 ;**
            ;2977 ; FUNCTIONAL DESCRIPTION:
            ;2978 ;
            ;2979 ; This routine is used to set the starting address of the
            ;2980 ; memory to the specified value.
            ;2981 ;
            ;2982 ; CALLING CONSTRAINT:
            ;2983 ;
            ;2984 ; =0
            ;2985 ;
            ;2986 ; INPUTS:
            ;2987 ;
            ;2988 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2989 ;     (B Register Image divided by 2**16)
            ;2990 ; rc[0e] - Contains Address of Register A
            ;2991 ;
            ;2992 ; OUTPUTS:
            ;2993 ;
            ;2994 ; d - Contains aligned starting address (B Register Image)
            ;2995 ;
            ;2996 ; SIDE EFFECTS:
            ;2997 ;
            ;2998 ; d,q,va, and sc are destroyed
            ;2999 ;--
            ;3000 SET.START.ADDR:
U 121D, 0010,0038,6580,0970,0284,721E  ;3001  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 121E, 0000,003C,01F8,0803,0000,121F  ;3002  va_va+4,q_0  ; Set address to Register B
            ;3003  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 121F, 0D00,003C,0980,0800,0084,B220  ;3004      sc_sc-k[.2]  ; Setup to insert bit 14
U 1220, 0801,001C,0180,0800,0000,1221  ;3005  d_d.ornot.mask  ; Insert Write Enable bit
U 1221, 0000,003E,0180,A800,0000,0001  ;3006  cache.p_d[long],return1 ; Set the starting address and return
            ;3007
```

```
            ;3008  .PAGE
            ;3009  .TOC "   ENABLE NEXT MS780-E"
            ;3010 ;++
            ;3011 ; FUNCTIONAL DESCRIPTION:
            ;3012 ;
            ;3013 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3014 ;    They will be tested  separately, according to the TR level at which
            ;3015 ;    they are located,  starting with the one  at the lowest level first.
            ;3016 ;          When  a test has finished  with the first unit, it will have to call
            ;3017 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3018 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3019 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3020 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3021 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3022 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3023 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3024 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3025 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3026 ;
            ;3027 ; CALLING CONSTRAINT:
            ;3028 ;
            ;3029 ;  =00
            ;3030 ;
            ;3031 ;--
            ;3032 ENABLE.NEXT.MS780E:
U 1222, 0000,003C,F9F0,2C00,0000,1223  ;3033  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3034  ALU_Q,    ; Check to see if it is zero
U 1223, 0001,203C,0180,0800,0010,1224  ;3035     CLK.UBCC  ; Check Condition Codes
U 1224, 0000,013C,0180,0800,0000,10E6  ;3036  Z?    ; ...for zero
U 10E6, 0000,003C,0180,0800,0000,1225  ;3037 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10E7, 0000,003E,0180,0800,0000,0002  ;3038  RETURN2    ; Zero No more units to test
            ;3039 ENABLE.NEXT:
U 1225, 0818,0038,C180,0800,0000,10E8  ;3040  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10E8, 0000,003D,0180,0800,0000,121D  ;3041 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10E9, 0818,0038,7980,0800,0000,1226  ;3042  D_K[.30]  ; Prepare to point to the ID register holding
            ;3043      ; ...the I/O Base address of the next MS780-E
U 1226, 0819,0014,1180,0800,0000,1227  ;3044  D_D+K[.4]  ; ...
U 1227, 0000,003C,F580,0800,0000,1228  ;3045  K[.A]    ; ...
U 1228, 0819,0014,F580,0800,0000,1229  ;3046  D_D+K[.A]  ; ...
U 1229, 0000,003C,F9F0,2C00,0000,122A  ;3047  Q_ID[3E]  ; Get MS780-E Counter
            ;3048  D_D-Q,    ; D now holds the pointer to the ID register
U 122A, 081D,0000,0180,0800,0082,122B  ;3049    SC_ALU  ; ...
U 122B, 0000,003C,01F0,2400,0000,122C  ;3050  Q_ID(SC)  ; Q gets adrress of next MS780E
U 122C, 0001,203C,0180,09F0,0000,122D  ;3051  RC[0E]_Q  ; Save it also in RC[0E]
U 122D, 0F03,003C,0180,09E8,0000,10EA  ;3052  RC[0D]_0,D_0  ; Set starting address to zero
U 10EA, 0000,003D,0180,0800,0000,121D  ;3053 =0 CALL[SET.START.ADDR] ; ....
U 10EB, 0F00,003C,F9F0,2C00,0000,122E  ;3054  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 122E, 081D,2008,0180,0800,0000,122F  ;3055  D_Q-D-1    ; Subtract 1 from total
U 122F, 0000,003C,F980,3C00,0000,10EC  ;3056  ID[3E]_D  ; Adjust the pointer
U 10EC, 0000,003D,0180,0800,0000,11D3  ;3057 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10ED, 0000,003E,0180,0800,0000,0001  ;3058  RETURN1    ; Tell caller another unit was found
            ;3059
```

```
            ;3060  .PAGE
            ;3061  .TOC "   GET MEMORY SIZE"
            ;3062 ;*************************************************************
            ;3063 ;++
            ;3064 ;
            ;3065 ; Functional Description:
            ;3066 ; ----------------------
            ;3067 ;
            ;3068 ; This subroutine can be used to get the size of the memory.
            ;3069 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3070 ; increments.
            ;3071 ;
            ;3072 ; Calling Constraint: =0
            ;3073 ; ------------------
            ;3074 ;
            ;3075 ;
            ;3076 ; Inputs:
            ;3077 ; ------
            ;3078 ;
            ;3079 ; RC[0E] Must have the Address of CNFG Register A
            ;3080 ;
            ;3081 ; Outputs:
            ;3082 ; -------
            ;3083 ;
            ;3084 ; Register D will contain upon exit the size of the
            ;3085 ; memory aligned on Mega-byte boundary.
            ;3086 ;
            ;3087 ; Side Effects:
            ;3088 ; ------------
            ;3089 ;
            ;3090 ; The contents of the following registers will be lost:
            ;3091 ;
            ;3092 ; D, SC, Q
            ;3093 ;
            ;3094 ;--
            ;3095 ;*************************************************************
            ;3096 GET.MEMORY.SIZE:
U 1230, 0010,0038,0180,0970,0200,1231  ;3097  VA_RC[0E]    ; Point to CNFG Reg A
U 1231, 0000,003C,0180,C800,0000,1232  ;3098  D[LONG]_CACHE.P    ; Read the register
U 1232, 001B,0000,D980,0800,0082,1233  ;3099  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 1233, 0D00,003C,0180,0800,0000,1234  ;3100  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 1234, 0819,0034,5580,0800,0000,1030  ;3101  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3102 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1030, 0000,003D,01E0,0800,0000,11C6  ;3103   Q_D    ; Save Memory size bits in Q
U 1031, 0000,003D,0D80,0800,0084,704C  ;3104  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3105       ; ...arrays on the controller
            ;3106 ;
            ;3107 ; The duplication of the next two return status is necessary so the returning
            ;3108 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3109 ; modifications.
            ;3110 ;
U 1032, 0819,2014,0580,0800,0000,1033  ;3111  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3112       ; ...(found 1 Meg arrays)
U 1033, 0819,2014,0580,0800,0000,1235  ;3113  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3114       ; ...(found 4 Meg arrays)
```

```
U 1235, 0000,003C,21F8,0800,0084,7236  ;3115  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
          ;3116  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 1236, 0D00,003E,0180,0800,0000,0001  ;3117   RETURN1   ; Return with memory size in Register D
          ;3118
```

```
            ;3119  .PAGE "   ERDAT ROUTINE"
            ;3120 ;
            ;3121 ;////////////////////////////////////////////////////////////////////////////
            ;3122 ;**
            ;3123 ;      THIS SUBROUTINE GETS THE ERROR DATA FROM ID[25-2X]
            ;3124 ; AND PUTS THAT DATA IN RC[0E]-RC[0X].
            ;3125 ;
            ;3126 ; (X DEPENDS ON HOW MANY ERRORS OCCURRED; MAXIMUM OF 6 DECIMAL)
            ;3127 ;
            ;3128 ; IT IS ASSUMMED THAT THE NUMBER OF ERRORS IS CONTAINED
            ;3129 ; IN THE STATE REGISTER.
            ;3130 ; (ALL F'S IS THE FLAG FOR END OF ERROR DATA).
            ;3131 ;--
            ;3132 ;////////////////////////////////////////////////////////////////////////////
            ;3133 ;
U 1237. 0000.003C.8580.0800.0184.7238  ;3134 ERDAT: FE&SC_K[.C]    ; INIT RC ADRS TO C
U 1238. 0000.003C.B9B0.0800.0084.9239  ;3135  SC_SC+K[.19]    ; INIT ID ADRS TO 25 (HEX)
U 1239. 0000.003C.01F0.2400.0000.123A  ;3136 ERDLP: Q_ID(SC)
U 123A. 0000.003C.0180.0800.0181.D23B  ;3137  FE_SC+1,SC_FE    ; INCR ID ADRS,SC_RC ADRS
U 123B. 0001.203C.01B0.0838.0000.123C  ;3138  RC(SC)_ALU,ALU_Q
U 123C. 0000.003C.0580.0800.0185.B23D  ;3139  SC_FE,FE_SC-K[.1]   ; SC_ID ADRS,FE_DECR'D RC ADRS
U 123D. 0000.003C.0580.0800.1414.B23E  ;3140  STATE_STATE-K[.1],CLK.UBCC ; DECR ERROR COUNTER
U 123E. 0000.123C.01B0.0800.0000.103B  ;3141  EALU.Z?     ; DONE LOOP ERROR # TIMES?
U 103B. 0000.003C.01B0.0800.0000.1239  ;3142 =1011 J/ERDLP     ; NOPE
U 103F. 0000.003C.01B0.0800.0081.123F  ;3143  SC_FE     ; PUT END FLAG IN RC[X]
            ;3144  RETURN1,    ; ...AND RETURN TO ERROR CALL.
U 123F. 001B.0D02.0580.083B.0000.0001  ;3145   RC(SC)_ALU,ALU_0-K[.1]
            ;3146
```

```
            ;3147  .PAGE
            ;3148  .TOC "   Set Diagnostic Mode Routine"
            ;3149 ;++
            ;3150 ; FUNCTIONAL DESCRIPTION:
            ;3151 ;
            ;3152 ; This routine is used to set the specified diagnostic mode
            ;3153 ; in Register B. Other bits in the register remain unchanged.
            ;3154 ;
            ;3155 ; CALLING CONSTRAINT:
            ;3156 ;
            ;3157 ; =0
            ;3158 ;
            ;3159 ; INPUTS:
            ;3160 ;
            ;3161 ; d - Contains the mode to set in bits<1:0>
            ;3162 ; rc[0e] - Contains base address of controller
            ;3163 ;
            ;3164 ; SIDE EFFECTS:
            ;3165 ;
            ;3166 ; d,va,sc are destroyed
            ;3167 ;--
            ;3168 SET_DIAG_MODE:
U 1240, 0010,0038,D9F8,0970,0284,7241  ;3169  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 1241, 0D00,003C,0180,0803,0000,1242  ;3170  va_va+4,d_dal.sc ; Shift specified mode into position
U 1242, 0000,003C,01E0,0800,0000,1243  ;3171  q_d    ; Save the diagnostic mode bits
U 1243, 0000,003C,0180,C800,0000,1244  ;3172  d[long]_cache.p  ; Read the contents of the CNFG register B
U 1244, 081D,0030,0180,0800,0000,1245  ;3173  d_d.or.q  ; Set the diagnostic mode bits
U 1245, 0000,003E,0180,A800,0000,0001  ;3174  cache.p_d[long],return1 ; Set the specified mode and return
            ;3175
```

```
;3176  .PAGE
;3177  .TOC "   Check for Odd or Even TR"
;3178 ;*************************************************************
;3179 ;++
;3180 ;
;3181 ; Functional Description:
;3182 ; ----------------------
;3183 ;
;3184 ;
;3185 ;      This subroutine will  determine  base  on  the  contents  of
;3186 ; RC[0E]  (which  contains  the  I/O  base register address for the
;3187 ; MS780-E/H currently under test), whether the memory is at Even or
;3188 ; Odd TR level.
;3189 ;
;3190 ;
;3191 ; Calling Constraint: =00
;3192 ; ------------------
;3193 ;
;3194 ;
;3195 ; Inputs:
;3196 ; ------
;3197 ;
;3198 ; RC[0E] must hold the I/O base address of the memory under test
;3199 ;
;3200 ;
;3201 ; Outputs:
;3202 ; -------
;3203 ;
;3204 ; There are no specific outputs for this routine.
;3205 ; The only output will be the RETURN mode:
;3206 ;
;3207 ;  RETURN1 for EVEN TR
;3208 ;
;3209 ;  RETURN2 for ODD TR
;3210 ;
;3211 ;
;3212 ; Side Effects:
;3213 ; ------------
;3214 ;
;3215 ; The contents of the following registers will be lost
;3216 ;
;3217 ; D, SC
;3218 ;
;3219 ;--
;3220 ;*************************************************************
;3221 ODD.EVEN.TR:
U 1246, 081B,001C,8580,0800,0000,1247  ;3222  D_NOT.K[.C]    ; Load D register with negative x12
U 1247, 0001,003C,0180,0800,0082,1248  ;3223  SC_D     ; Prepare SC for right shift of x12
U 1248, 0810,0038,0180,0970,0000,1249  ;3224  D_RC[0E]    ; Get the I/O base address
U 1249, 0000,003C,01F8,0800,0000,124A  ;3225  Q_0     ; Clear Q Reg
U 124A, 0D00,003C,0180,0800,0000,124B  ;3226  D_DAL.SC    ; Shift D right 13 bits
U 124B, 0000,193C,0180,0800,0000,106E  ;3227  D1-0?     ; Is bit 0 of D set?
U 106E, 0000,003E,0180,0800,0000,0001  ;3228 =1110 RETURN1     ; EVEN TR
U 106F, 0000,003E,0180,0800,0000,0002  ;3229  RETURN2     ; ODD TR
;3230
```

```
        :3231
```

```
;3232  .PAGE
;3233  .TOC "   BOARD ADDRESS GENERATOR"
;3234 ;*************************************************************
;3235 ;++
;3236 ;
;3237 ; Functional Description:
;3238 ; ----------------------
;3239 ;
;3240 ;
;3241 ;      This subroutine will be used by board  addressing  tests  to
;3242 ; generate  an  address  for  location  0  of  a board based on the
;3243 ; following parameters passed by the calling routine:
;3244 ;
;3245 ;
;3246 ; (RC[08])  -  Pointer to the board currently unde test
;3247 ;
;3248 ; (RC[09])  -  Binary Flag indicating the type of memory chips
;3249 ;   and Interleave Mode
;3250 ;
;3251 ;  RC[09] Bit 1  0
;3252 ;  --------------
;3253 ;
;3254 ;  0  0  -  64K chips Non-Interleaved
;3255 ;
;3256 ;  0  1  -  256K chips Non-Interleaved
;3257 ;
;3258 ;  1  0  -  64K chips Ext-Interleaved
;3259 ;
;3260 ;  1  1  -  256K chips Ext-Interleaved
;3261 ;
;3262 ;
;3263 ; (RC[0A])  -  Should be 0 if memory is at odd TR or
;3264 ;   x8 if memory is at even TR
;3265 ;
;3266 ;
;3267 ;
;3268 ; Calling Constraint: =0
;3269 ; ------------------
;3270 ;
;3271 ;
;3272 ; Inputs:
;3273 ; ------
;3274 ;
;3275 ; See above description
;3276 ;
;3277 ;
;3278 ; Outputs:
;3279 ; -------
;3280 ;
;3281 ; D register will hold the BOARD address
;3282 ;
;3283 ;
;3284 ; Side Effects:
;3285 ; ------------
;3286 ;
```

```
            ;3287 ; The contents of the following registers will be lost:
            ;3288 ;
            ;3289 ; D, Q, SC
            ;3290 ;
            ;3291 ;--
            ;3292 ;*****************************************************************
            ;3293 GEN.BOARD.ADD:
U 124C, 0810,0038,0180,0948,0000,124D  ;3294  D_RC[9]     ; Get FLAG
U 124D, 0000,003C,01F8,0800,0000,124E  ;3295   Q_0    ; Clear Q Reg
U 124E, 0000,003C,0580,0800,0084,724F  ;3296  SC_K[.1]    ; Prepare the SC for the shift
U 124F, 0000,193C,0180,0800,0000,107C  ;3297  D1-0?     ; See what configuration we have
U 107C, 0000,003C,0180,0800,0000,1250  ;3298 =1100 J/BOARD.64K.NINTR   ; 64K chips Non-Interleaved
U 107D, 0000,003C,0180,0800,0000,1250  ;3299  J/BOARD.256K.NINTR   ; 256K chips Non-Interleaved
U 107E, 0000,003C,0180,0800,0000,1251  ;3300  J/BOARD.64K.INTR   ; 64K Interleaved
U 107F, 0000,003C,0180,0800,0000,1251  ;3301  J/BOARD.256K.INTR   ; 256K Interleaved
            ;3302
            ;3303 BOARD.64K.NINTR:
            ;3304 BOARD.256K.NINTR:
            ;3305  D_RC[8],    ; Set the Board address
U 1250, 0810,003A,0180,0940,0000,0001  ;3306   RETURN1    ; ...and return to caller
            ;3307 BOARD.64K.INTR:
            ;3308 BOARD.256K.INTR:
U 1251, 0810,0038,0180,0940,0000,1252  ;3309  D_RC[8]     ; Get Board address
U 1252, 0D00,003C,0180,0800,0000,1253  ;3310  D_DAL.SC    ; Shift left the number of bits
            ;3311      ; ...specified by the SC
            ;3312  D_D[A+B]RC[0A],    ; Add TR displacement and
U 1253, 0811,0016,0180,0950,0000,0001  ;3313   RETURN1    ; ...return to the caller
            ;3314
```

```
;3315  .PAGE
;3316  .TOC "   BANK ADDRESS GENEREATOR"
;3317 ;*************************************************************
;3318 ;**
;3319 ;
;3320 ; Functional Description:
;3321 ; ---------------------
;3322 ;
;3323 ;
;3324 ;      This subroutine will be used by the bank addressing tests to
;3325 ; generate an address.  This address will be generated based on the
;3326 ; following parameters passed from the calling routine:
;3327 ;
;3328 ;
;3329 ; (RC[08])  -  Pointer to the board on which the bank under test
;3330 ;   will be found
;3331 ;
;3332 ; (RC[09])  -  Binary Flag indicating the type of memory chips
;3333 ;   and Interleave Mode
;3334 ;
;3335 ;  RC[09] Bit 1  0
;3336 ;  --------------
;3337 ;
;3338 ;  0  0  -  64K chips Non-Interleaved
;3339 ;
;3340 ;  0  1  -  256K chips Non-Interleaved
;3341 ;
;3342 ;  1  0  -  64K chips Ext-Interleaved
;3343 ;
;3344 ;  1  1  -  256K chips Ext-Interleaved
;3345 ;
;3346 ;
;3347 ; (RC[0A])  -  Should be 0 if memory is at odd TR or
;3348 ;   x8 if memory is at even TR
;3349 ;
;3350 ; (RC[0B])  -  Number of the bank under test (from 0 to 4)
;3351 ;
;3352 ;
;3353 ;
;3354 ; Calling Constraint: =0
;3355 ; ------------------
;3356 ;
;3357 ;
;3358 ; Inputs:
;3359 ; ------
;3360 ;
;3361 ; See description above for RC[8] through RC[0B]
;3362 ;
;3363 ;
;3364 ; Outputs:
;3365 ; -------
;3366 ;
;3367 ; When this subroutine is exitted Register D
;3368 ; will hold the bank address
;3369 ;
```

```
            ;3370 ; Side Effects:
            ;3371 ; ------------
            ;3372 ;
            ;3373 ; The contents of the following registers will be lost:
            ;3374 ;
            ;3375 ; D, Q, SC
            ;3376 ;
            ;3377 ;--
            ;3378 ;*************************************************************
            ;3379 =0
            ;3380 GEN.BANK.ADD:
U 10EE, 0000,003D,0180,0800,0000,124C  ;3381  CALL[GEN.BOARD.ADD]  ; Generate first the address of
            ;3382       ; ...the board
U 10EF, 0000,003C,01E0,0800,0000,1254  ;3383  Q_D    ; Save the BOARD address
U 1254, 0000,003C,2180,0800,0084,7255  ;3384  SC_K[.14]   ; Get
U 1255, 0810,0038,0180,0948,0000,1256  ;3385  D_RC[9]    ; Get FLAG
U 1256, 0000,193C,0180,0800,0000,108C  ;3386  DI-0?    ; See which configuration we have
            ;3387 =1100 J/BANK.64K.NINTR,  ; 64K chips Non-Interleaved
U 108C, 0810,0038,0180,0958,0000,1257  ;3388   D_RC[0B]  ; Get BANK Pointer
            ;3389  J/BANK.256K.NINTR,  ; 256K chips Non-Interleaved
U 108D, 0810,0038,0180,0958,0000,1257  ;3390   D_RC[0B]  ; Get BANK Pointer
U 108E, 0000,003C,0180,0800,0000,125A  ;3391  J/BANK.64K.INTR   ; 64K chips Interleaved
            ;3392       ; ...SC is Pointing to bit 20
            ;3393  J/BANK.256K.INTR,  ; 256K chips Interleaved
U 108F, 0000,003C,0980,0800,0084,925A  ;3394   SC_SC+K[.2]  ; ...point SC to bit 22
            ;3395 ;
            ;3396 ;
            ;3397 BANK.64K.NINTR:
            ;3398 BANK.256K.NINTR:
U 1257, 0819,0034,0D80,0800,0000,1258  ;3399  D_D.AND.K[.3]   ; Mask all other bits except <1:0>
U 1258, 0100,003C,0180,0800,0000,1259  ;3400  D_D.LEFT2   ; Shift BANK Pointer into position
            ;3401  D_D.OR.Q,   ; Insert BANK address in BOARD address
U 1259, 081D,0032,0180,0800,0000,0001  ;3402   RETURN1   ; Return to caller
            ;3403 BANK.64K.INTR:
            ;3404 BANK.256K.INTR:
U 125A, 0810,0038,0180,0958,0000,125B  ;3405  D_RC[0B]   ; Get BANK Pointer
            ;3406  ALU_D.AND.K[.1],  ; See if bit 0 is set
U 125B, 0019,0034,0580,0800,0010,125C  ;3407   CLK.UBCC  ; ...
U 125C, 0000,013C,0180,0800,0000,10F0  ;3408  Z?
U 10F0, 0019,2030,11C0,0800,0000,10F1  ;3409 =0 Q_Q.OR.K[.4]   ; Bit 0 was set in BANK Pointer
            ;3410       ; ...so set bit 2 in BANK address
            ;3411  ALU_D.AND.K[.2],  ; See if bit 1 is set in BANK Pointer
U 10F1, 0019,0034,0980,0800,0010,125D  ;3412   CLK.UBCC  ; ...
U 125D, 0000,013C,0180,0800,0000,10F2  ;3413  Z?
            ;3414 =0 Q_Q.ORNOT.MASK,   ; Bit 1 was set in BANK Pointer
U 10F2, 0001,201C,01C0,0800,0000,125E  ;3415   J/BANK.EXIT  ; ...so set bit pointed by SC in
            ;3416       ; ...BANK address
U 10F3, 0001,2034,01C0,0800,0000,125E  ;3417  Q_Q.AND.MASK   ; Bit 1 was NOT set in BANK Pointer
            ;3418       ; ...so make sure that bit pointed
            ;3419       ; ...by SC in BANK address is clear
            ;3420 BANK.EXIT:
            ;3421  D_Q,    ; Save BANK Address in D
U 125E, 0C00,003E,0180,0800,0000,0001  ;3422   RETURN1   ; ...return to caller
            ;3423
            ;3424
```

```
;3425  .PAGE
;3426  .TOC "   REED MULLER ENCODING ROUTINE"
;3427 ;*****************************************************************
;3428 ;++
;3429 ;
;3430 ; Functional Description:
;3431 ; ---------------------
;3432 ;
;3433 ;      This subroutine is used to apply the Reed-Muller
;3434 ; encoding  algorithm to a 16 bit  vector with an error
;3435 ; handling capability of up to 3 bits per 32 bit vector.
;3436 ;
;3437 ;
;3438 ; NOTE: The DESIGN SPEC for the MS780-E/H diagnostic
;3439 ;  includes a theoretical discussion of the
;3440 ;  Reed-Muller codes as well as the algorithm
;3441 ;  for the following subroutines:
;3442 ;
;3443 ;  ENCODE
;3444 ;
;3445 ;  DECODE
;3446 ;
;3447 ;  FIRST.ORDER
;3448 ;
;3449 ;  SECOND.ORDER
;3450 ;
;3451 ;
;3452 ; Calling Constraint: =0
;3453 ; ------------------
;3454 ;
;3455 ;
;3456 ; Inputs:
;3457 ; ------
;3458 ;
;3459 ; RC[4] must contain the 16 bit vector that should
;3460 ; be encoded.
;3461 ;
;3462 ; Outputs:
;3463 ; -------
;3464 ;
;3465 ; RC[1] will contain upon exit the encoded vector
;3466 ;
;3467 ;
;3468 ; Side Effects:
;3469 ; ------------
;3470 ;
;3471 ; The contents of the following registers will be lost:
;3472 ;
;3473 ; R[0] through R[0B], RC[1], T1 through T6
;3474 ;
;3475 ;--
;3476 ;*****************************************************************
;3477 ENCODE:
U 125F, 0018,0038,0580,0AD0,0000,1260  ;3478  R[0A]_K[.1]   ; Initialize SYMBOL_POINTER (to the
;3479       ; ...bits of the INFO_SYMBOLS vector)
```

```
U 1260, 0003,003C,0180,0988,0000,1261  ;3480  RC[1]_0     ; Clear CODE_VECTOR location
U 1261, 0810,0038,0180,0920,0000,1262  ;3481  D_RC[4]     ; Get INFO_SYMBOLS to be coded
U 1262, 0001,003C,0180,0990,0000,1263  ;3482  RC[2]_D     ; Initialize INFO_SYMBOLS
        ;3483  ALU_D[AND]R[0A],  ; See if BIT 0 of INFO_SYMBOLS is 0
U 1263, 000D,0034,0180,0A50,0010,1264  ;3484   CLK.UBCC  ; ...
U 1264, 0800,013C,0180,0A50,0000,10F4  ;3485  Z?,D_R[0A]   ; Get SYMBOL_POINTER
U 10F4, 001B,001C,1980,0988,0000,10F5  ;3486 =0 RC[1]_NOT.K[ZERO]  ; BIT 0 was NOT 0, set CODE_VECTOR to
        ;3487       ; ...all ones
U 10F5, 0021,003C,0180,0AD0,0000,10F6  ;3488  R[0A]_D.LEFT,SI/ZERO  ; Shift SYMBOL_POINTER left one bit
        ;3489       ; ...pointing to the next bit to
        ;3490       ; ...be encoded
U 10F6, 0000,003D,0180,0800,0000,1267  ;3491 =0 CALL[FIRST.ORDER]  ; Start encoding the first order
        ;3492       ; ...symbols into the final vector
U 10F7, 0000,003C,0180,0800,0000,10F8  ;3493  NOP
U 10F8, 0000,003D,0180,0800,0000,1273  ;3494 =0 CALL[SECOND.ORDER]  ; Start encoding the second order
        ;3495       ; ...symbols into the final vector
U 10F9, 0000,003E,0180,0800,0000,0001  ;3496  RETURN1    ; Return to caller with the encoded
        ;3497       ; ...vector in RC[1]
        ;3498
        ;3499
        ;3500
```

```
            ;3501  .PAGE
            ;3502  .TOC "   REED-MULLER VECTOR GENERATION ROUTINE"
            ;3503 ;*****************************************************************
            ;3504 ;*
            ;3505 ; THIS SUBROUTINE GENERATES THE VECTORS USED BY THE
            ;3506 ; REED-MULLER ALGORITHM TO GENERATE CODEWORDS.  THESE
            ;3507 ; VECTORS ARE:
            ;3508 ;
            ;3509 ;  V1: 10101010101010101010101010101010
            ;3510 ;  V2: 11001100110011001100110011001100
            ;3511 ;  V3: 11110000111100001111000011110000
            ;3512 ;  V4: 11111111000000001111111100000000
            ;3513 ;  V5: 11111111111111110000000000000000
            ;3514 ;
            ;3515 ; THE ROUTINE IS ARRANGED IN A TABLE, IN ORDER TO
            ;3516 ; BRANCH INTO IT WITH D3-0? BRANCH.
            ;3517 ;
            ;3518 ; **NOTE**: D<3:0> MUST HAVE THE VECTOR NUMBER WHEN
            ;3519 ;    CALLING THE ROUTINE.
            ;3520 ;
            ;3521 ; THE VECTOR IS RETURNED IN THE D-REGISTER.
            ;3522 ;
            ;3523 ; SIDE EFFECTS:
            ;3524 ;
            ;3525 ; The contents of the following registers will be lost:
            ;3526 ;
            ;3527 ; Q, SC
            ;3528 ;-
            ;3529 GENERATE.VECTOR:
U 1265, 0500,003C,0180,0800,0000,1266  ;3530  D_D.LEFT,SI/ZERO  ; Multiply VECTOR Pointer by 2
            ;3531       ; ... so as to index into the table
            ;3532       ; ...below
U 1266, 0000,193C,0180,0800,0000,1090  ;3533  D3-0?    ; Index into the Vector table
            ;3534 =0000 D_NOT.K[ZERO],   ; Generate V(0)
U 1090, 081B,001E,1980,0800,0000,0001  ;3535   RETURN1   ; ...
U 1091, 0000,003C,0180,0800,0000,1092  ;3536  NOP
U 1092, 001B,0000,1180,0800,0082,1093  ;3537 V1: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1093, 0F1B,0010,11C0,0800,0000,109C  ;3538  ALU_0+K[.4]+1,Q_ALU,D_0,J/VSHF  ; Q<3:0>=0101
U 1094, 001B,0000,1180,0800,0082,1095  ;3539 V2: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1095, 0F1B,0038,0DC0,0800,0000,109C  ;3540  Q_K[.3],D_0,J/VSHF  ; Q<3:0>=0011
U 1096, 001B,0000,0180,0800,0082,1097  ;3541 V3: ALU_0-K[.8],SC_ALU  ; SET SHIFT VALUE = -8
U 1097, 0F1B,0038,61C0,0800,0000,109C  ;3542  Q_K[.F],D_0,J/VSHF  ; Q<7:0>=00001111
U 1098, 001B,001C,6580,0800,0082,1099  ;3543 V4: ALU_NOT.K[.10],SC_ALU  ; K[SLOW],SHIFT VALUE = -16.
U 1099, 0F1B,0038,49C0,0800,0080,D09C  ;3544  Q_K[.FF],D_0,SC_SC+1,J/VSHF  ; Q<7:0>=11111111
U 109A, 081B,0038,C180,0800,0000,109F  ;3545 V5: D_K[.FFFF],J/V.EXIT
U 109B, 0000,003C,0180,0800,0000,109C  ;3546  NOP
U 109C, 0D00,003C,0180,0800,0000,109D  ;3547 VSHF: D_DAL,SC
U 109D, 0000,193C,0180,0800,0000,109E  ;3548  D0?    ; ANY V(I) IS DONE WHEN D<0>=1
U 109E, 0000,003C,0180,0800,0000,109C  ;3549 =1110 J/VSHF    ; LUCKY BR CONSTRAINT
            ;3550 V.EXIT:
            ;3551  D_NOT.D,   ; Adjust D as necessary
U 109F, 0B01,002A,0180,0800,0000,0001  ;3552   RETURN1   ; Return to caller
            ;3553
```

```
            ;3554  .PAGE
            ;3555  .TOC "    FIRST ORDER REED-MULLER ENCODE"
            ;3556 ;*************************************************************
            ;3557 ;**
            ;3558 ;
            ;3559 ; Functional Description:
            ;3560 ; ----------------------
            ;3561 ;
            ;3562 ;      This subroutine is used by the ENCODE and DECODE
            ;3563 ; routines to encode the first order Reed-Muller vector
            ;3564 ; bits <05:01> of the received vector.
            ;3565 ;
            ;3566 ;
            ;3567 ; Calling Constraint: =0
            ;3568 ; ------------------
            ;3569 ;
            ;3570 ;
            ;3571 ; Inputs:
            ;3572 ; ------
            ;3573 ;
            ;3574 ; RC[2] must hold the vector to be encoded (INFO_SYMBOLS)
            ;3575 ;
            ;3576 ; R[0A] must hold the SYMBOL_POINTER as passed by
            ;3577 ; subroutines ENCODE or DECODE
            ;3578 ;
            ;3579 ; RC[1] must hold the CODE_VECTOR as passed by
            ;3580 ; subroutines ENCODE or DECODE
            ;3581 ;
            ;3582 ; Outputs:
            ;3583 ; -------
            ;3584 ;
            ;3585 ; RC[1] will contain upon exit the CODE_VECTOR for
            ;3586 ; bits <05:01> of INFO_SYMBOLS
            ;3587 ;
            ;3588 ; R[0A] will have an UPDATED SYMBOL_POINTER
            ;3589 ;
            ;3590 ;
            ;3591 ; Side Effects:
            ;3592 ; ------------
            ;3593 ;
            ;3594 ; The contents of the following registers will be lost
            ;3595 ;
            ;3596 ; D, Q, R[0], R[2]
            ;3597 ;
            ;3598 ;--
            ;3599 ;*************************************************************
            ;3600 FIRST.ORDER:
U 1267, 0018,0038,0980,0A80,0000,1268  ;3601  R[0]_K[.2]    ; Initialize the COUNTER
U 1268, 0018,0038,0580,0A90,0000,1269  ;3602  R[2]_K[.1]    ; Initialize variable J
            ;3603 FRST.ORDR.LOOP:
U 1269, 0810,0038,0180,0910,0000,126A  ;3604  D_RC[2]     ; Get INFO_SYMBOLS
            ;3605  ALU_D[AND]R[0A],  ; See if bit pointed by SYMBOL_POINTER
U 126A, 000D,0034,0180,0A50,0010,126B  ;3606   CLK.UBCC  ; ...in INFO_SYMBOLS is set
U 126B, 0000,013C,0180,0800,0000,10FA  ;3607  Z?
U 10FA, 0000,003C,0180,0800,0000,10FC  ;3608 =0 J/FRST.ORDR.SYM.NOT0  ; Bit was not 0
```

```
U 10FB, 0000,003C,0180,0800,0000,126D  ;3609  J/FRST.ORDR.SYM.EQ0  ; Bit was 0
          ;3610 =0
          ;3611 FRST.ORDR.SYM.NOT0:
          ;3612  D_R[2],    ; Point to VECTOR(J)
U 10FC, 0800,003D,0180,0A10,0000,1265  ;3613  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 10FD, 0010,0038,01C0,0908,0000,126C  ;3614  Q_RC[1]    ; Get CODE_VECTOR
          ;3615  ALU_D.XOR.Q,    ; Encode current bit pointed by
U 126C, 001D,0020,0180,0988,0000,126D  ;3616   RC[1]_ALU  ; ...SYMBOL_POINTER and save in RC[1]
          ;3617 FRST.ORDR.SYM.EQ0:
U 126D, 0800,003C,0180,0A10,0000,126E  ;3618  D_R[2]    ; Get J
U 126E, 0019,0014,0580,0A90,0000,126F  ;3619  R[2]_D+K[.1]   ; Increment J
U 126F, 0800,003C,0180,0A50,0000,1270  ;3620  D_R[0A]    ; Get SYMBOL_POINTER
U 1270, 0021,003C,D580,0AD0,0000,1271  ;3621  R[0A]_D.LEFT,SI/ZERO,K[.6] ; Point to the next bit to be encoded
          ;3622  ALU_R[0][XOR]K[.6],   ; Is the COUNTER equal to 6?
U 1271, 0018,0020,D580,0A00,0010,1272  ;3623   CLK.UBCC  ; ...(i.e., COUNTER .LE. 5)
          ;3624  Z?,    ;
U 1272, 0800,013C,0180,0A00,0000,10FE  ;3625   D_R[0]   ; Get the COUNTER just in case it has
          ;3626       ; ...to be incremented
          ;3627 =0 J/FRST.ORDR.LOOP,  ; Counter is NOT equal to 6 yet, thus
U 10FE, 0019,0014,0580,0A80,0000,1269  ;3628   R[0]_D+K[.1]  ; ...we did not encode all First
          ;3629       ; ...Order Symbols.  Repeat the loop
U 10FF, 0000,003E,0180,0800,0000,0001  ;3630  RETURN1    ; OK. COUNTER is equal to 6 so
          ;3631       ; ...we finished encoding ALL First
          ;3632       ; ...Order Symbols.  Return to caller.
          ;3633
          ;3634
```

```
                ;3635  .PAGE
                ;3636  .TOC "    SECOND ORDER REED-MULLER ENCODING ROUTINE"
                ;3637 ;*************************************************************
                ;3638 ;++
                ;3639 ;
                ;3640 ; Functional Description:
                ;3641 ; ----------------------
                ;3642 ;
                ;3643 ;      This subroutine is used by the ENCODE and DECODE
                ;3644 ; routines to encode the Second order Reed-Muller vector
                ;3645 ; of INFO_SYMBOLS vector.
                ;3646 ;
                ;3647 ;
                ;3648 ; Calling Constraint: =0
                ;3649 ; ------------------
                ;3650 ;
                ;3651 ;
                ;3652 ; Inputs:
                ;3653 ; ------
                ;3654 ;
                ;3655 ; RC[1] must hold the CODE_VECTOR as set up by FIRST.ORDER
                ;3656 ; routine or by the DECODE routine.
                ;3657 ;
                ;3658 ; RC[2] must hold the INFO_SYMBOLS vector (to be encoded)
                ;3659 ;
                ;3660 ; R[0A] must hold the SYMBOL_POINTER as set up by FIRST.ORDER
                ;3661 ; routine or by the DECODE routine.
                ;3662 ;
                ;3663 ;
                ;3664 ; Outputs:
                ;3665 ; -------
                ;3666 ;
                ;3667 ; RC[1] will hold upon exit the encoded vector
                ;3668 ;
                ;3669 ;
                ;3670 ; Side Effects:
                ;3671 ; ------------
                ;3672 ;
                ;3673 ; The contents of the following registers will be lost:
                ;3674 ;
                ;3675 ; D, Q, RC[1], R[0] through R[2], R[0A], SC
                ;3676 ;
                ;3677 ;--
                ;3678 ;*************************************************************
                ;3679 SECOND.ORDER:
U 1273, 0018.0038.0580.0A80.0000.1274  ;3680  R[0]_K[.1]    ; Initialize COUNTER to 1
U 1274, 0F18.0038.1180.0A90.0000.1275  ;3681  R[2]_K[.4],D_0    ; Initialize variable J to 4
U 1275, 0019.0010.1180.0A88.0000.1276  ;3682  ALU_D+K[.4]+1,R[1]_ALU  ; Initialize variable I to 5
                ;3683 SCND.ORDR.LOOP:
U 1276, 0810.0038.0180.0910.0000.1277  ;3684  D_RC[2]     ; Get INFO_SYMBOLS
                ;3685  ALU_D[AND]R[0A],  ; See if bit pointed by SYMBOL_POINTER
U 1277, 000D.0034.0180.0A50.0010.1278  ;3686   CLK.UBCC  ; ...in INFO_SYMBOLS is set
U 1278, 0000.013C.0180.0800.0000.110A  ;3687  Z?
U 110A, 0000.003C.0180.0800.0000.1110  ;3688 =0 J/SCND.ORDR.SYM.NOT0  ; Bit was not 0
U 110B, 0000.003C.0180.0800.0000.127B  ;3689  J/SCND.ORDR.SYM.EQ0  ; Bit was 0
```

```
            ;3690 =0
            ;3691 SCND.ORDR.SYM.NOT0:
            ;3692  D_R[1],     ; Point to VECTOR(I)
U 1110. 0800.003D.0180.0A08.0000.1265  ;3693  CALL[GENERATE.VECTOR]  ; ...and call routine to
U 1111. 0001.003C.0180.0AF0.0000.1112  ;3694  R[0E]_D     ; Save VECTOR(I)
            ;3695 =0 D_R[2],     ; Point to VECTOR(J)
U 1112. 0800.003D.0180.0A10.0000.1265  ;3696  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 1113. 0000.003C.01C0.0A70.0000.1279  ;3697  Q_R[0E]     ; Get saved VECTOR(I)
            ;3698  D_D.AND.Q.    ; "Combine" VECTOR(I) with VECTOR(J)
U 1279. 081D.0034.0180.0908.0000.127A  ;3699   LC_RC[1]   ; Get CODE_VECTOR
U 127A. 0011.0020.0180.0988.0000.127B  ;3700  RC[1]_D.XOR.LC    ; Encode bit pointed by SYMBOL_POINTER
            ;3701       ; ...in CODE_VECTOR
            ;3702 SCND.ORDR.SYM.EQ0:
            ;3703  ALU_R[2][A-B]K[.1],D_ALU, ; Decrement J
U 127B. 0818.0000.0580.0A10.0010.127C  ;3704   CLK.UBCC  ; Is it 0 yet?
            ;3705  Z?,
U 127C. 0000.013C.01C0.0A08.0000.1114  ;3706   Q_R[1]    ; Get I
            ;3707 =0 J/SCND.ORDR.JNT0,  ; J is NOT 0.
U 1114. 0001.003C.0180.0A90.0000.127E  ;3708   R[2]_D    ; Save J
U 1115. 0019.2000.0580.0A90.0000.127D  ;3709  R[2]_Q-K[.1]    ; Decrement I and save
U 127D. 0019.2000.0580.0A88.0000.127E  ;3710  R[1]_Q-K[.1]    ; Reinitialize J to I - 1
            ;3711 SCND.ORDR.JNT0:
U 127E. 0800.003C.0180.0A50.0000.127F  ;3712  D_R[0A]     ; Get SYMBOL_POINTER
U 127F. 0021.003C.F5F8.0AD0.0000.1280  ;3713  R[0A]_D.LEFT,SI/ZERO,Q_0,K[.A] ; Point to the next bit to be encoded
U 1280. 0819.2010.F580.0800.0000.1281  ;3714  ALU_Q+K[.A]+1,D_ALU  ; D = d11
            ;3715  ALU_D.XOR.R[0],    ; Is the COUNTER equal to d11?
U 1281. 000D.0020.0180.0A00.0010.1282  ;3716   CLK.UBCC  ; ...
            ;3717  Z?,     ; ...
U 1282. 0800.013C.0180.0A00.0000.1116  ;3718   D_R[0]    ; Get the COUNTER in case it is
            ;3719       ; ...not equal to d11
            ;3720 =0 J/SCND.ORDR.LOOP,  ; COUNTER is NOT equal to d11 yet
U 1116. 0019.0014.0580.0A80.0000.1276  ;3721   R[0]_D+K[.1]  ; Increment and save the COUNTER
U 1117. 0000.003E.0180.0800.0000.0001  ;3722  RETURN1     ; Return to caller.
            ;3723       ; ...Finished encoding all second
            ;3724       ; ...order terms
            ;3725
```

```
                ;3726  .PAGE
                ;3727  .TOC "   REED-MULLER DECODE ROUTINE"
                ;3728 ;*****************************************************************
                ;3729 ;**
                ;3730 ;
                ;3731 ; Functional Description:
                ;3732 ; ----------------------
                ;3733 ;
                ;3734 ;      This subroutine is used to decode the Reed-Muller
                ;3735 ; codes encoded by the ENCODE routine.
                ;3736 ;
                ;3737 ; Calling Constraint: =00
                ;3738 ; ------------------
                ;3739 ;
                ;3740 ;
                ;3741 ; Inputs:
                ;3742 ; ------
                ;3743 ;
                ;3744 ; RC[0] must hold the 32 bit vector to be decoded
                ;3745 ;
                ;3746 ;
                ;3747 ; Outputs:
                ;3748 ; -------
                ;3749 ;
                ;3750 ; RC[2] will hold the decoded vector
                ;3751 ;
                ;3752 ;
                ;3753 ; Side Effects:
                ;3754 ; ------------
                ;3755 ;
                ;3756 ;
                ;3757 ; See header of subroutine ENCODE
                ;3758 ;
                ;3759 ;
                ;3760 ;--
                ;3761 ;*****************************************************************
                ;3762 DECODE:
U 1283, 0018,0038,6580,0A98,0000,1284  ;3763  R[3]_K[.10]    ; Initialize variable K to d16
U 1284, 0018,0038,6580,0AA0,0000,1285  ;3764  R[4]_K[.10]    ; Initialize variable L to d16
U 1285, 0003,003C,0180,0990,0000,1286  ;3765  RC[2]_0     ; Initialize variable INFO_SYMBOLS to 0
U 1286, 0810,0038,0180,0900,0000,1287  ;3766  D_RC[0]     ; Get the RCVD_VECTOR
U 1287, 0001,003C,0180,0AB0,0000,1288  ;3767  R[6]_D     ; Save RCVD_VECTOR
                ;3768 DECODE.SCND.ORDR.LOOP:
U 1288, 0840,003C,0180,0A20,0000,1289  ;3769  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; Divide L by 2
U 1289, 0001,003C,0180,0AA0,0000,128A  ;3770  R[4]_D     ; L = L/2
U 128A, 0003,003C,0180,0AC0,0000,128B  ;3771  R[8]_0     ; Initialize variable 1ST to 0
U 128B, 0003,003C,0180,0AC8,0000,128C  ;3772  R[9]_0     ; Initialize variable SAVE_1ST to 0
U 128C, 0003,003C,0180,0998,0000,128D  ;3773  RC[3]_0     ; Initialize varaible COVERED_BITS to 0
U 128D, 0018,0038,0580,0A80,0000,128E  ;3774  R[0]_K[.1]    ; Initialize COUNTER to 1
U 128E, 0003,003C,0180,0AB8,0000,128F  ;3775  R[7]_0     ; Initialize variable ONE_COUNTER to 0
                ;3776 DECODE.SEARCH.LOOP:
U 128F, 0003,003C,0180,0AA8,0000,1290  ;3777  R[5]_0     ; Initialize variable SUM to 0
U 1290, 0003,003C,0180,0AD8,0000,1291  ;3778  R[0B]_0     ; Initialize variable CURRENT_BIT_MASK
                ;3779       ; ...to 0
U 1291, 0000,003C,0180,0A40,0082,1292  ;3780  SC_R[8]     ; Get 1ST
```

```
U 1292, 0003,001C,0180,0AD8,0000,1293  ;3781  R[0B]_NOT.MASK   ; Set bit 1ST in CURRENT_BIT_MASK
          ;3782  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST set in RCVD_VECTOR?
U 1293, 0000,0024,0180,0A30,0010,1294  ;3783   CLK.UBCC  ; ...
U 1294, 0800,013C,0180,0A28,0000,1118  ;3784  Z?,D_R[5]   ; Get SUM
U 1118, 0019,0014,0580,0AA8,0000,1119  ;3785 =0 R[5]_D+K[.1]   ; Bit 1ST in RCVD_VECTOR was NOT 0,
          ;3786       ; ...increment the SUM
U 1119, 0800,003C,0180,0A40,0000,1295  ;3787  D_R[8]    ; Get 1ST
U 1295, 000D,0014,01C0,0A20,0082,1296  ;3788  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 1296, 0800,001C,0180,0A58,0000,1297  ;3789  D_R[0B].ORNOT.MASK  ; Set bit 1ST + L in CURRENT_BIT_MASK
U 1297, 0001,003C,0180,0AD8,0000,1298  ;3790  R[0B]_D    ; Save CURRENT_BIT_MASK
          ;3791  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + L set in RCVD_VECTOR?
U 1298, 0000,0024,0180,0A30,0010,1299  ;3792   CLK.UBCC  ; ...
U 1299, 0800,013C,0180,0A28,0000,111A  ;3793  Z?,D_R[5]   ; Get SUM
U 111A, 0019,0014,0580,0AA8,0000,111B  ;3794 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 111B, 0800,003C,0180,0A58,0000,129A  ;3795  D_R[0B]    ; Get CURRENT_BIT_MASK
U 129A, 000D,2014,0180,0A18,0082,129B  ;3796  ALU_Q[A+B]R[3],SC_ALU   ; SC = 1ST + K + L
U 129B, 0001,001C,0180,0AD8,0000,129C  ;3797  R[0B]_D.ORNOT.MASK  ; Set bit 1ST + K + L in
          ;3798       ; ...CURRENT_BIT_MASK
          ;3799  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + K + L set
U 129C, 0000,0024,0180,0A30,0010,129D  ;3800   CLK.UBCC  ; ...in RCVD_VECTOR?
U 129D, 0800,013C,0180,0A28,0000,111C  ;3801  Z?,D_R[5]   ; Get SUM
U 111C, 0019,0014,0580,0AA8,0000,111D  ;3802 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 111D, 0800,003C,0180,0A40,0000,129E  ;3803  D_R[8]    ; Get 1ST
U 129E, 000D,0014,0180,0A18,0082,129F  ;3804  ALU_D[A+B]R[3],SC_ALU   ; SC = 1ST + K
U 129F, 0800,001C,0180,0A58,0000,12A0  ;3805  D_R[0B].ORNOT.MASK  ; Set bit 1ST + K in CURRENT_BIT_MASK
U 12A0, 0001,003C,0180,0AD8,0000,12A1  ;3806  R[0B]_D    ; Save CURRENT_BIT_MASK
          ;3807  ALU_R[6].ANDNOT.MASK,  ; Is bit 1ST + K set in RCVD_VECTOR
U 12A1, 0000,0024,0180,0A30,0010,12A2  ;3808   CLK.UBCC  ; ...
U 12A2, 0800,013C,0180,0A28,0000,111E  ;3809  Z?,D_R[5]   ; Get SUM
U 111E, 0019,0014,0580,0AA8,0000,111F  ;3810 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 111F, 0800,003C,0180,0A58,0000,12A3  ;3811  D_R[0B]    ; Get CURRENT_BIT_MASK
          ;3812  ALU_D[AND]RC[3],  ; Are bits set in CURRENT_BIT_MASK
U 12A3, 0011,0034,0180,0918,0010,12A4  ;3813   CLK.UBCC  ; ...also set in COVERED_BITS
U 12A4, 0000,013C,0180,0800,0000,1120  ;3814  Z?
          ;3815 =0 J/DECODE.BITS.COVERED,  ; Some of the bits in CURRENT_BIT_MASK
U 1120, 0820,003C,0180,0A20,0000,12A5  ;3816  D_R[4].LEFT,SI/ZERO  ; ...have been previously covered.
          ;3817       ; ...D = 2 * L
          ;3818  J/DECODE.BITS.NOT.COVERED, ; Bits in CURRENT_BIT_MASK were not
U 1121, 0818,0014,0580,0A40,0000,12A8  ;3819  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
          ;3820 DECODE.BITS.COVERED:
U 12A5, 080D,0014,0180,0A48,0000,12A6  ;3821  ALU_D[A+B]R[9],D_ALU   ; D = SAVE_1ST + (2 * L)
U 12A6, 0001,003C,0180,0AC0,0000,12A7  ;3822  R[8]_D    ; 1ST = SAVE_1ST + (2 * L)
          ;3823  R[9]_D,     ; SAVE_1ST = SAVE_1ST + (2 * L)
U 12A7, 0001,003C,0180,0AC8,0000,128F  ;3824   J/DECODE.SEARCH.LOOP ; Repeat the search loop for
          ;3825       ; ...unCOVERED_BITS
          ;3826 DECODE.BITS.NOT.COVERED:
U 12A8, 0001,003C,0180,0AC0,0000,12A9  ;3827  R[8]_D    ; 1ST = 1ST + 1
U 12A9, 0800,003C,0180,0A58,0000,12AB  ;3828  D_R[0B]    ; Get CURRENT_BIT_MASK
U 12AB, 0811,0030,0180,0918,0000,12AC  ;3829  D_D.OR.RC[3]   ; COVERED_BITS = COVERED_BITS .OR.
          ;3830       ; ...CURRENT_BIT_MASK
U 12AC, 0001,003C,0180,0998,0000,12AD  ;3831  RC[3]_D    ; Save COVERED_BITS mask
U 12AD, 0000,003C,01C0,0A38,0000,12AE  ;3832  Q_R[7]    ; Get ONE_COUNTER
U 12AE, 0800,003C,0180,0A28,0000,12AF  ;3833  D_R[5]    ; Get SUM and see if it is EVEN or ODD
U 12AF, 0000,193C,0180,0800,0000,10AE  ;3834  D1-0?    ; ...
U 10AE, 0000,003C,0180,0800,0000,12B0  ;3835 =1110 J/DECODE.SUM.EVEN  ; SUM was even
```

```
U 10AF, 0019,2014,0580,0AB8,0000,12B0  ;3836  R[7]_Q+K[.1]   ; SUM is ODD.  Increment ONE_COUNTER
        ;3837 DECODE.SUM.EVEN:
        ;3838  ALU_R[0].XOR.K[.8],  ; Is COUNTER equal to 8?
U 12B0, 0018,0020,0180,0A00,0010,12B1  ;3839   CLK.UBCC  ; ...
U 12B1, 0800,013C,0180,0A00,0000,1122  ;3840  Z?,D_R[0]   ; Get COUNTER in case it has to be
        ;3841       ; ...incremented
        ;3842 =0 J/DECODE.SEARCH.LOOP,  ; COUNTER is NOT equal to 8 yet
U 1122, 0019,0014,0580,0A80,0000,128F  ;3843   R[0]_D+K[.1]  ; Increment COUNTER
        ;3844  ALU_R[7][A-B]K[.4],  ; Is ONE_COUNTER GREATER than or
U 1123, 0018,0000,1180,0A38,0010,12B2  ;3845   CLK.UBCC  ; ...EQUAL to  4?
        ;3846  ALU.N?,    ; Check Condition Codes.
        ;3847   D_RC[2],  ; Get INFO_SYMBOLS
U 12B2, 0810,1B38,4580,0910,0000,1047  ;3848   K[.8000]  ; Bring up slow constant
U 1047, 0819,0030,4580,0800,0000,104F  ;3849 =0111 D_D.OR.K[.8000]   ; ONE_COUNTER GREATER than or
        ;3850       ; ...EQUAL to 4.  Set bit 0 of
        ;3851       ; ...INFO_SYMBOLS
        ;3852  RC[2]_ALU.RIGHT,SI/ZERO, ; Shift INFO_SYMBOLS right and Save
U 104F, 0041,003C,0180,0990,0000,12B3  ;3853   ALU_D   ; ...
        ;3854  ALU_R[4][XOR]K[.1],  ; Is L EQUAL to 1?
U 12B3, 0018,0020,0580,0A20,0010,12B4  ;3855   CLK.UBCC  ; ...
        ;3856  Z?,
U 12B4, 0840,013C,0180,0A18,0000,1124  ;3857  D_ALU.RIGHT,ALU_R[3],SI/ZERO ; D = K/2. In case L = 1
U 1124, 0000,003C,0180,0800,0000,1288  ;3858 =0 J/DECODE.SCND.ORDR.LOOP  ; L was not EQUAL to 1. Not all SECOND
        ;3859       ; ...ORDER symbols have been decoded.
        ;3860       ; ...Repeat loop.
        ;3861  ALU_D.XOR.K[.1],  ; Is K = 1?
U 1125, 0019,0020,0580,0800,0010,12B5  ;3862   CLK.UBCC  ; ...
U 12B5, 0001,013C,0180,0A98,0000,1126  ;3863  Z?,R[3]_D   ; Save K just in case.
        ;3864 =0 J/DECODE.SCND.ORDR.LOOP, ; OK. K was updated and it is NOT equal
U 1126, 0001,003C,0180,0AA0,0000,1288  ;3865   R[4]_D   ; ...to 1.  Update L.  L = K/2
U 1127, 0810,0038,0180,0910,0000,12B6  ;3866  D_RC[2]    ; Get INFO_SYMBOLS
U 12B6, 0200,003C,0180,0800,0000,12B7  ;3867  D_D.RIGHT2,SI/ZERO  ; Shift INFO_SYMBOLS right 4 bits
U 12B7, 0200,003C,0180,0800,0000,12B8  ;3868  D_D.RIGHT2,SI/ZERO  ; ...
U 12B8, 0001,003C,0180,0990,0000,12B9  ;3869  RC[2]_D    ; Save INFO_SYMBOLS
U 12B9, 0018,0038,0980,0AD0,0000,12BA  ;3870  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 12BA, 0003,003C,0180,0988,0000,1128  ;3871  RC[1]_0    ; Clear CODE_VECTOR
U 1128, 0000,003D,0180,0800,0000,1273  ;3872 =0 CALL[SECOND.ORDER]  ; Re-encode the second order symbols
        ;3873       ; ...so as to extract them from
        ;3874       ; ...RCVD_VECTOR
U 1129, 0800,003C,0180,0A30,0000,12BB  ;3875  D_R[6]    ; Get RCVD_VECTOR
U 12BB, 0811,0020,0180,0908,0000,12BC  ;3876  D_D.XOR.RC[1]   ; "Subtract" the encoded second order
        ;3877       ; ...symbols from RCVD_VECTOR
U 12BC, 0001,003C,0180,0AB0,0000,12BD  ;3878  R[6]_D    ; Save the result
U 12BD, 0018,0038,7580,0AA0,0000,12BE  ;3879  R[4]_K[.20]   ; Initialize variable L to d32
        ;3880 DECODE.FRST.ORDER.LOOP:
U 12BE, 0840,003C,0180,0A20,0000,12BF  ;3881  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; D = L/2
U 12BF, 0001,003C,0180,0AA0,0000,12C0  ;3882  R[4]_D    ; Save L
U 12C0, 0003,003C,0180,0AC0,0000,12C1  ;3883  R[8]_0    ; Initialize variable 1ST
U 12C1, 0003,003C,0180,0AC8,0000,12C2  ;3884  R[9]_0    ; Initialize variable SAVE_1ST
U 12C2, 0003,003C,0180,0998,0000,12C3  ;3885  RC[3]_0    ; Initialize variable COVERED_BITS to 0
U 12C3, 0018,0038,0580,0A80,0000,12C4  ;3886  R[0]_K[.1]   ; Initialize COUNTER to 1
U 12C4, 0003,003C,0180,0AB8,0000,12C5  ;3887  R[7]_0    ; Clear ONE_COUNTER
        ;3888 DECODE.FRST.ORDR.SEARCH:
U 12C5, 0003,003C,0180,0AA8,0000,12C6  ;3889  R[5]_0    ; Initialize variable SUM to 0
U 12C6, 0003,003C,0180,0AD8,0000,12C7  ;3890  R[0B]_0    ; Initialize variable CURRENT_BIT_MASK
```

```
            ;3891       ; ...to 0
U 12C7, 0000,003C,0180,0A40,0082,12C8  ;3892  SC_R[8]     ; Get 1ST
U 12C8, 0003,001C,0180,0AD8,0000,12C9  ;3893  R[0B]_NOT.MASK    ; Set bit 1ST in CURRENT_BIT_MASK
            ;3894  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST set in RCVD_VECTOR?
U 12C9, 0000,0024,0180,0A30,0010,12CA  ;3895   CLK.UBCC   ; ...
U 12CA, 0800,013C,0180,0A28,0000,112A  ;3896  Z?,D_R[5]    ; Get SUM
U 112A, 0019,0014,0580,0AA8,0000,112B  ;3897 =0 R[5]_D+K[.1]    ; Bit 1ST in RCVD_VECTOR was NOT 0,
            ;3898       ; ...increment the SUM
U 112B, 0800,003C,0180,0A40,0000,12CB  ;3899  D_R[8]     ; Get 1ST
U 12CB, 000D,0014,01C0,0A20,0082,12CC  ;3900  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 12CC, 0800,001C,0180,0A58,0000,12CD  ;3901  D_R[0B].ORNOT.MASK   ; Set bit 1ST + L in CURRENT_BIT_MASK
U 12CD, 0001,003C,0180,0AD8,0000,12CE  ;3902  R[0B]_D     ; Save CURRENT_BIT_MASK
            ;3903  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + L set in RCVD_VECTOR?
U 12CE, 0000,0024,0180,0A30,0010,12CF  ;3904   CLK.UBCC   ; ...
U 12CF, 0800,013C,0180,0A28,0000,112C  ;3905  Z?,D_R[5]    ; Get SUM
U 112C, 0019,0014,0580,0AA8,0000,112D  ;3906 =0 R[5]_D+K[.1]    ; Bit was NOT 0. Increment SUM
U 112D, 0800,003C,0180,0A58,0000,12D0  ;3907  D_R[0B]     ; Get CURRENT_BIT_MASK
            ;3908  ALU_D[AND]RC[3],    ; Are bits set in CURRENT_BIT_MASK
U 12D0, 0011,0034,0180,0918,0010,12D1  ;3909   CLK.UBCC   ; ...also set in COVERED_BITS
U 12D1, 0000,013C,0180,0800,0000,112E  ;3910  Z?
            ;3911 =0 J/DECODE.BITS.COVRD,   ; Some of the bits in CURRENT_BIT_MASK
U 112E, 0820,003C,0180,0A20,0000,12D2  ;3912  D_R[4].LEFT,SI/ZERO   ; ...have been previously covered.
            ;3913       ; ...D = 2 * L
            ;3914  J/DECODE.BITS.NOT.COVRD, ; Bits in CURRENT_BIT_MASK were not
U 112F, 0818,0014,0580,0A40,0000,12D5  ;3915  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
            ;3916 DECODE.BITS.COVRD:
U 12D2, 080D,0014,0180,0A48,0000,12D3  ;3917  ALU_D[A+B]R[9],D_ALU   ; D = SAVE_1ST + (2 * L)
U 12D3, 0001,003C,0180,0AC0,0000,12D4  ;3918  R[8]_D     ; 1ST = SAVE_1ST + (2 * L)
            ;3919  R[9]_D,     ; SAVE_1ST = SAVE_1ST + (2 * L)
U 12D4, 0001,003C,0180,0AC8,0000,12C5  ;3920     J/DECODE.FRST.ORDR.SEARCH ; Repeat the search loop for
            ;3921       ; ...unCOVERED_BITS
            ;3922 DECODE.BITS.NOT.COVRD:
U 12D5, 0001,003C,0180,0AC0,0000,12D6  ;3923  R[8]_D     ; 1ST = 1ST + 1
U 12D6, 0800,003C,0180,0A58,0000,12D7  ;3924  D_R[0B]     ; Get CURRENT_BIT_MASK
U 12D7, 0811,0030,0180,0918,0000,12D8  ;3925  D_D.OR.RC[3]    ; COVERED_BITS = COVERED_BITS .OR.
            ;3926       ; ...CURRENT_BIT_MASK
U 12D8, 0001,003C,0180,0998,0000,12D9  ;3927  RC[3]_D     ; Save COVERED_BITS mask
U 12D9, 0000,003C,01C0,0A38,0000,12DA  ;3928  Q_R[7]     ; Get ONE_COUNTER
U 12DA, 0800,003C,0180,0A28,0000,12DB  ;3929  D_R[5]     ; Get SUM and see if it is EVEN or ODD
U 12DB, 0000,193C,0180,0800,0000,10BE  ;3930  D[-0?     ; ...
U 10BE, 0000,003C,0180,0800,0000,12DC  ;3931 =1110 J/DECODE.SUM.EVEN2   ; SUM was even
U 10BF, 0019,2014,0580,0AB8,0000,12DC  ;3932  R[7]_Q+K[.1]    ; SUM is ODD.  Increment ONE_COUNTER
            ;3933 DECODE.SUM.EVEN2:
            ;3934  ALU_R[0].XOR.K[.10],   ; Is COUNTER equal to d16?
U 12DC, 0018,0020,6580,0A00,0010,12DD  ;3935   CLK.UBCC   ; ...
U 12DD, 0800,013C,0180,0A00,0000,1130  ;3936  Z?,D_R[0]    ; Get COUNTER in case it has to be
            ;3937       ; ...incremented
            ;3938 =0 J/DECODE.FRST.ORDR.SEARCH, ; COUNTER is NOT equal to 16 yet
U 1130, 0019,0014,0580,0A80,0000,12C5  ;3939    R[0]_D+K[.1]   ; Increment COUNTER
            ;3940  ALU_R[7][A-B]K[.8],   ; Is ONE_COUNTER GREATER than or
U 1131, 0018,0000,0180,0A38,0010,12DE  ;3941    CLK.UBCC   ; ...EQUAL to  8?
            ;3942  ALU.N?,     ; Check Condition Codes.
U 12DE, 0810,1B38,0180,0910,0000,1057  ;3943    D_RC[2]    ; Get INFO_SYMBOLS
U 1057, 0819,0030,0580,0800,0000,105F  ;3944 =0111 D_D.OR.K[.1]    ; ONE_COUNTER GREATER than or
            ;3945       ; ...EQUAL to 8.  Set bit 0 of
```

```
            ;3946       ; ...INFO_SYMBOLS
U 105F, 0021,003C,0180,0990,0000,12DF  ;3947  RC[2]_D.LEFT,SI/ZERO  ; Shift INFO_SYMBOLS left and Save
            ;3948  ALU_R[4].XOR.K[.1],  ; Is L = 1?
U 12DF, 0018,0020,0580,0A20,0010,12E0  ;3949   CLK.UBCC
U 12E0, 0000,013C,0180,0800,0000,1132  ;3950  Z?
U 1132, 0000,003C,0180,0800,0000,12BE  ;3951 =0 J/DECODE.FRST.ORDER.LOOP ; L is NOT equal to 1 yet. Repeat loop.
U 1133, 0003,003C,0180,0988,0000,12E1  ;3952  RC[1]_0     ; Initialize CODE_VECTOR to 0
U 12E1, 0018,0038,0980,0AD0,0000,1134  ;3953  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 1134, 0000,003D,0180,0800,0000,1267  ;3954 =0 CALL[FIRST.ORDER]  ; Generate the first order Reed-Muller
            ;3955       ; ...code for the symbols just
            ;3956       ; ...decoded
U 1135, 0800,003C,0180,0A30,0000,12E2  ;3957  D_R[6]     ; Get RCVD_VECTOR
U 12E2, 0010,0038,01C0,0908,0000,12E3  ;3958  Q_RC[1]     ; Get CODE_VECTOR
U 12E3, 081D,2020,0180,0AB0,0000,12E4  ;3959  Q_Q.XOR.D,R[6]_ALU  ; RCVD_VECTOR = RCVD_VECTOR .XOR.
            ;3960       ; ... CODE_VECTOR
U 12E4, 0003,003C,0180,0AB8,0000,12E5  ;3961  R[7]_0     ; Initialize ONE_COUNTER to 0
U 12E5, 0018,0038,0580,0AD0,0000,12E6  ;3962  R[0A]_K[.1]   ; Set Bit 0 of SYMBOL_POINTER
            ;3963 DECODE.ZERO.ORDR.LOOP:
U 12E6, 0800,003C,0180,0A50,0000,12E7  ;3964  D_R[0A]     ; Get SYMBOL_POINTER
            ;3965  ALU_D[AND]R[6],    ; Is SYMBOL_POINTER .AND. RCVD_VECTOR
U 12E7, 000D,0034,0180,0A30,0010,12E8  ;3966   CLK.UBCC  ; ...equal to 0?
            ;3967  Z?,
U 12E8, 0800,013C,0180,0A38,0000,1136  ;3968   D_R[7]    ; Get ONE_COUNTER
            ;3969 =0 J/DECODE.ZERO.ORDR.BITNZ, ; Bit was NOT 0 in RCVD_VECTOR
U 1136, 0019,0014,0580,0AB8,0000,12E9  ;3970   R[7]_D+K[.1]  ; Increment ONE_COUNTER
            ;3971  J/DECODE.ZERO.ORDR.BITZ, ; Bit was equal to 0
U 1137, 0800,003C,0180,0A50,0000,12EC  ;3972   D_R[0A]    ; Get SYMBOL_POINTER
            ;3973 DECODE.ZERO.ORDR.BITNZ:
U 12E9, 0800,003C,0180,0A30,0000,12EA  ;3974  D_R[6]     ; Get RCVD_VECTOR
            ;3975  ALU_D[ANDNOT]R[0A],D_ALU, ; RCVD_VECTOR = RCVD_VECTOR .ANDNOT.
U 12EA, 080D,0024,0180,0A50,0010,12EB  ;3976   CLK.UBCC  ; ...SYMBOL_POINTER (i.e., clear bit
            ;3977       ; ...tested and see if the overall
            ;3978       ; ...result is 0)
U 12EB, 0000,013C,0180,0800,0000,1138  ;3979  Z?
            ;3980 =0 J/DECODE.ZERO.ORDR.BITZ, ; RCVD_VECTOR is NOT zero so
U 1138, 0800,003C,0180,0A50,0000,12EC  ;3981   D_R[0A]    ; ...continue decoding all zero
            ;3982       ; ...order bits
            ;3983  J/DECODE.ZERO.ORDR.DONE, ; RCVD_VECTOR is 0 so we finished
U 1139, 0800,003C,0180,0A38,0000,12EE  ;3984   D_R[7]    ; ...decoding all zero order bits.
            ;3985 DECODE.ZERO.ORDR.BITZ:
            ;3986  R[0A]_D.LEFT,SI/ZERO,   ; Arithmetic Shift Left SYMBOL_POINTER.
U 12EC, 0021,003C,0180,0AD0,0010,12ED  ;3987   CLK.UBCC  ; ...and see if it is 0
U 12ED, 0000,013C,0180,0800,0000,113A  ;3988  Z?
U 113A, 0000,003C,0180,0800,0000,12E6  ;3989 =0 J/DECODE.ZERO.ORDR.LOOP   ; ...Continue decoding
U 113B, 0800,003C,0180,0A38,0000,12EE  ;3990  D_R[7]     ; Get ONE_COUNTER
            ;3991 DECODE.ZERO.ORDR.DONE:
            ;3992  ALU_D-K[.10],    ; Is ONE_COUNTER GREATER than or EQUAL
U 12EE, 0019,0000,6580,0800,0010,12EF  ;3993   CLK.UBCC  ; ...to d16.
            ;3994  ALU.N?,
U 12EF, 0810,1B38,0180,0910,0000,10C7  ;3995   D_RC[2]    ; Get INFO_SYMBOLS
U 10C7, 0019,0030,0580,0990,0000,10CF  ;3996 =0111 RC[2]_D.OR.K[.1]  ; Set Bit 0 of INFO_SYMBOLS
U 10CF, 0000,003C,0180,0800,0000,12F0  ;3997  NOP
            ;3998 ;
            ;3999 ;-
            ;4000 ;
```

```
            ;4001 ; This is where the actual decoding of the received Reed-Muller
            ;4002 ; code (RCVD_VECTOR) ends.  The decoded result is left in
            ;4003 ; RC[2]  (INFO_SYMBOLS) which will be now compared against
            ;4004 ; the expected value which is in RC[4].
            ;4005 ;
            ;4006 ;++
            ;4007 ;
U 12F0, 0810,0038,0180,0910,0000,12F1  ;4008  D_RC[2]     ; Get INFO_SYMBOLS
            ;4009  ALU_D[XOR]RC[4],   ; See if INFO_SYMBOLS is equal to
U 12F1, 0011,0020,0180,0920,0010,12F2  ;4010   CLK.UBCC   ; ...the expected data
U 12F2, 0000,013C,0180,0800,0000,113C  ;4011  Z?
            ;4012 =0 J/DECODE.ERROR,   ; Data Returned is not equal to
U 113C, 0000,003C,0180,0800,1480,12F3  ;4013  SC_STATE    ; ...the expected.  Get number of
            ;4014       ; ...errors
U 113D, 0000,003E,0180,0800,0000,0003  ;4015  RETURN3     ; OK. No error in data so do a
            ;4016       ; ...RETURN3
            ;4017 DECODE.ERROR:
U 12F3, 0000,003C,E980,0800,0084,92F4  ;4018  SC_SC+K[.24]    ; Get number of errors and transform it
            ;4019       ; ...in an ID Register address
            ;4020       ; ...starting with ID register 25
U 12F4, 0000,003C,0580,0800,0084,92F5  ;4021  SC_SC+K[.1]    ; SC Reg holds a pointer to ID regs 25
            ;4022       ; ...through 2X depending on how many
            ;4023       ; ...errors occurred.
U 12F5, 0810,0038,0180,0910,0000,12F6  ;4024  D_RC[2]     ; Get Received DECODED data
U 12F6, 0000,003C,0180,3400,0000,12F7  ;4025  ID(SC)_D    ; Save it in ID Reg pointed by SC
            ;4026  D_RC[4],    ; Get Expected Data
U 12F7, 0810,0038,0580,0920,0084,92F8  ;4027   SC_SC+K[.1]  ; Increment pointer to ID Regs
U 12F8, 0000,003C,0180,3400,0000,12F9  ;4028  ID(SC)_D    ; Save Expected data in ID reg
U 12F9, 0000,003C,0980,0800,1404,92FA  ;4029  STATE_STATE+K[.2]  ; Increment the error counter.
U 12FA, 0000,003C,0180,0800,1480,12FB  ;4030  SC_STATE    ; Get the number of errors
U 12FB, 0818,0038,1D80,0800,0000,12FC  ;4031  D_SC     ; ...in D reg
            ;4032  ALU_D.XOR.K[.C],   ; Is the error counter equal to
U 12FC, 0019,0020,8580,0800,0010,12FD  ;4033   CLK.UBCC   ; ...x0C? (i.e., maximum number of
            ;4034       ; ...errors has been reached)
U 12FD, 0000,013C,0180,0800,0000,113E  ;4035  Z?
U 113E, 0000,003E,0180,0800,0000,0002  ;4036 =0 RETURN2     ; Maximum number of errors has NOT been
            ;4037       ; ...reached yet
U 113F, 0000,003E,0180,0800,0000,0001  ;4038  RETURN1     ; Maximum number of errors!!
            ;4039
            ;4040
            ;4041
```

```
;4042  .PAGE
;4043  .TOC "   FIND CONTROLLERS"
;4044 ;*************************************************************
;4045 ;**
;4046 ;
;4047 ; Functional Description:
;4048 ; ----------------------
;4049 ;
;4050 ;
;4051 ;      This subroutine is called by tests that  need  to  know  how
;4052 ; many  controllers  are available on the MS780-E/H currently under
;4053 ; test.  It makes  use  of  subroutine  SELECT.CNTRLR  to  put  the
;4054 ; controllers in different Interleave Modes.
;4055 ; The subroutine returns control to the calling routine  by  either
;4056 ; one  of:   Return1, Return2, Return3 or Return4.  It is up to the
;4057 ; test to decode this information as necessary.
;4058 ; Since some Tests require this feature, the subroutine also  finds
;4059 ; the  memory  size  available  on each controller and leaves it in
;4060 ; registers R[0] and R[1].
;4061 ;
;4062 ;
;4063 ; Calling Constraint: =000
;4064 ; ------------------
;4065 ;
;4066 ;
;4067 ; Inputs:
;4068 ; ------
;4069 ;
;4070 ; RC[0E] MUST HOLD THE I/O BASES ADDRESS OF THE MS780-E/H
;4071 ; UNDER TEST
;4072 ;
;4073 ; Outputs:
;4074 ; -------
;4075 ;
;4076 ; - Return1, Return2, Return3 or Return4 depending on
;4077 ;   how many Controllers were found
;4078 ;
;4079 ; - R[0] Size of Memory on the Lower Controller
;4080 ;   R[1] Size of Memory on the Upper Controller
;4081 ;
;4082 ; Side Effects:
;4083 ; ------------
;4084 ;
;4085 ; The contents of the following registers will be lost:
;4086 ;
;4087 ; D, Q, SC, R[0], R[1]
;4088 ;
;4089 ;--
;4090 ;*************************************************************
;4091 =00
;4092 FIND.CNTRLRS:
;4093  D_K[ZERO],   ; Attempt to select the Lower
U 1034, 0818,0039,1980,0800,0000,11BC  ;4094   CALL[SELECT.CNTRLR] ; ...Controller
U 1035, 0000,003C,0180,0800,0000,1048  ;4095  J/LOW.MIS.X    ; Lower Controller Misconfigured
U 1036, 0000,003D,0180,0800,0000,1230  ;4096  CALL[GET.MEMORY.SIZE]  ; Lower Controller has been  configured
```

```
            ;4097       ; ...find its memory size
U 1037, 0001,003C,0180,0A80,0000,1040  ;4098  R[0]_D     ; Save Memory size in R[0]
            ;4099 =00 D_K[.2],    ; Attempt to select the Upper
U 1040, 0818,0039,0980,0800,0000,11BC  ;4100   CALL[SELECT.CNTRLR] ; ...Controller
U 1041, 0000,003E,0180,0800,0000,0002  ;4101  RETURN2     ; Upper Controller Misconfigured.
            ;4102       ; ...Let caller know that ONLY the
            ;4103       ; ...Lower Controller was found
U 1042, 0000,003D,0180,0800,0000,1230  ;4104  CALL[GET.MEMORY.SIZE]  ; Upper Controller was Configured so
            ;4105       ; ...find its memory size
            ;4106  R[1]_D,     ; Save Memory Size in R[1]
U 1043, 0001,003E,0180,0A88,0000,0004  ;4107   RETURN4    ; Let Caller know that BOTH Controllers
            ;4108       ; ...were found
            ;4109 =00
            ;4110 LOW.MIS.X:
            ;4111  D_K[.2],    ; Lower Controller was not found,
U 1048, 0818,0039,0980,0800,0000,11BC  ;4112   CALL[SELECT.CNTRLR] ; ...so attempt to Configure the
            ;4113       ; ...Upper Controller
U 1049, 0000,003E,0180,0800,0000,0001  ;4114  RETURN1     ; Tell Caller that NONE of the
            ;4115       ; ...Controllers were found
U 104A, 0000,003D,0180,0800,0000,1230  ;4116  CALL[GET.MEMORY.SIZE]  ; Upper Controller was found,
            ;4117       ; ...go get its memory size
            ;4118  R[1]_D,     ; Save the memory size in R[1]
U 104B, 0001,003E,0180,0A88,0000,0003  ;4119   RETURN3    ; Let Caller know that ONLY the
            ;4120       ; ...Upper Controller was found
            ;4121
            ;4122
            ;4123
```

```
            ;4124  .PAGE
            ;4125  .TOC "   DELAY"
            ;4126 ;*****************************************************************
            ;4127 ;**
            ;4128 ;
            ;4129 ; Functional Description:
            ;4130 ; ----------------------
            ;4131 ;
            ;4132 ; This subroutine is used to reset the Time-Out counter in
            ;4133 ; Mic-Mon.  This may be desired when a test is running a loop
            ;4134 ; and does not have to do any calls to Mic-Mon.
            ;4135 ;
            ;4136 ;
            ;4137 ; Calling Constraint: =0
            ;4138 ; ------------------
            ;4139 ;
            ;4140 ;
            ;4141 ; Inputs:
            ;4142 ; ------
            ;4143 ;
            ;4144 ; NONE
            ;4145 ;
            ;4146 ;
            ;4147 ; Outputs:
            ;4148 ; -------
            ;4149 ;
            ;4150 ; Time-Out counter in Mic-Mon is Reset
            ;4151 ;
            ;4152 ;
            ;4153 ; Side Effects:
            ;4154 ; ------------
            ;4155 ;
            ;4156 ; Contents of the D reg are lost
            ;4157 ;
            ;4158 ;
            ;4159 ;
            ;4160 ;--
            ;4161 ;*****************************************************************
U 12FE, 0818,0038,0180,0800,0000,105E  ;4162 DELAY: D_K[.8],J/CALLCON
            ;4163
            ;4164
```

```
;4165   .PAGE "TEST 1B9 BOARD/BANK SELECT ADDRESSING"
;4166  ;*************************************************************
;4167  ;+**
;4168  ;
;4169  ; Functional Description:
;4170  ; ----------------------
;4171  ;
;4172  ;       The following subtests check the addressing lines to the boards on
;4173  ;   each controllers and to the banks on each board.
;4174  ;
;4175  ;       SUBTEST 1 - BOARD SELECT ADDRESSING (Non-Intrlvd)
;4176  ;
;4177  ;      This subtest finds the number of array boards present on each controller
;4178  ; In Non-Interleaved mode, it makes a first pass starting with board 0, writing
;4179  ; encoded 0's  to  location 0  on each board (up to the last existing board).##
;4180  ; Then, the  test  starts  from  board 1  reading out the  data of 0 previously
;4181  ; written.  If it checks out  (i.e., no errors)  a unique  encoded data pattern
;4182  ; is written to address 0 of each board.  Board 0 address location 0 is written
;4183  ; last.  The data is  read out,  starting  with board 0, decoded and checked to
;4184  ; see if matches the expected data (i.e.,  the correct board was addressed both
;4185  ; by reading and writing).
;4186  ;
;4187  ;       SUBTEST 2 - BOARD SELECT ADDRESSING (Ext-Intrlvd)
;4188  ;
;4189  ;       This subtest performs exactly the same test as the previous one, except
;4190  ;  that  in  this case the  controller  under test is placed in the Externally
;4191  ;  Interleaved mode.
;4192  ;
;4193  ;      SUBTEST 3 - BANK SELECT ADDRESSING (Non-Intrlvd)
;4194  ;
;4195  ;      In this subtest the bank select addressing lines are being checked in the
;4196  ;    Non-Interleaved mode.  It starts by writing encoded zeroes to all the banks
;4197  ;    on an array board. Then starting with bank 0 unique encoded data is written
;4198  ;    to location 0 on each bank  after first  checking and making sure that the
;4199  ;    data previously written is still valid.##
;4200  ;    Now, starting with bank 0 the data is retreived, decoded and checked to see
;4201  ;    if it is equal to the expected.
;4202  ;
;4203  ;       SUBTEST 4 - BANK SELECT ADDRESSING (Ext-Intrlvd)
;4204  ;
;4205  ;       This is a repeat of the previous subtest except that the test is
;4206  ;      performed in the externally interleaved mode.
;4207  ;
;4208  ;    The above tests had to be repeated in interleaved and Non-Interleaved Modes
;4209  ;   due to the changes in addressing that take place when changing from one mode
;4210  ;   to the next.
;4211  ;
;4212  ;
;4213  ;
;4214  ;
;4215  ; Logic Description:
;4216  ; -----------------
;4217  ;         Failure of these tests identify problems with the addressing logic on
;4218  ;    the SBI Interface board, the address latches (i.e., RAM Address Queue and
;4219  ; Latch,and Board/Bank Address Queue and Latch) on the controller board, the
```

```
;4220 ; array address drivers or the physical lines connecting them.
;4221 ;
;4222 ;      Please (!) NOTE: in the case of the bank addressing tests the banks will
;4223 ;  be  "assigned"  consecutive  numbers  from 1  to  some  maximum  value. This
;4224 ;  maximum   value  is  dependent  on the  number  of  arrays  present  on  the
;4225 ;  controller.## Thus if there were 3 arrays on one controller (upper or lower)
;4226 ;  then the  bank  numbers  will be from 1 to 12, each bank on each array being
;4227 ;  assigned  a unique number.
;4228 ;
;4229 ;       Bank numbers 1 through 4 will be on the first array, banks 5 through 8
;4230 ;    will be  on the  second array and finally banks 9 through 12 will be on the
;4231 ;    third array.##  This numbering scheme is necessary to differentiate between
;4232 ;    the boards in the case of a failure. (A better way would have been to print
;4233 ;    the board number along  with the failing banks on that board.  However this
;4234 ;    is impossible due to  the limited number of registers.)
;4235 ;
;4236 ;
;4237 ; Error Logout:
;4238 ; ------------
;4239 ;
;4240 ; See individual error reports.
;4241 ;
;4242 ; Sync Point Description:
;4243 ; ----------------------
;4244 ;  N/A
;4245 ;
;4246 ;=================================================================
;4247 ;
;4248 ; Register Map:
;4249 ; ------------
;4250 ;
;4251 ;      RA,RB      Description:
;4252 ;      -----      -----------
;4253 ;
;4254 ;  NOT USED
;4255 ;
;4256 ;=================================================================
;4257 ;
;4258 ;       RC        Description:
;4259 ;       --        -----------
;4260 ; E MS780-E BASE ADRRES
;4261 ;
;4262 ; D CNFG C/D ADRRES OF CURRENTLY TESTED MS780-E
;4263 ;
;4264 ; C UPPER/LOWER FLAG (1= UPPER CONTROLLER, 0= LOWER CONTROLLER)
;4265 ;
;4266 ; 4 PATTERN COUNTER
;4267 ;
;4268 ; 6 MAX ADDRESS+1
;4269 ;
;4270 ; 7 DISPLACEMENT
;4271 ;
;4272 ; 8  BOARD POINTER
;4273 ;
;4274 ; 9 INTERLEAVED MODE BITS
```

```
            ;4275 ;
            ;4276 ; A  CONTROLLER SELECT ACCORDING TO TR (0=EVEN, 8=ODD)
            ;4277 ;
            ;4278 ;       B  ADDRESS REFERANCE OR BANK COUNTER SEE INDIVIDUAL SUBTESTS
            ;4279 ;
            ;4280 ;--
            ;4281 ;=============================================================================
            ;4282 =0
U 1140, 0000,003D,0180,0800,0000,1087  ;4283 TST.30: NEWTST
U 1141, 0000,003C,0180,0800,0000,1050  ;4284  NOP
            ;4285 =00
U 1050, 0000,003D,0180,0800,0000,10D0  ;4286  CALL[FIND.MS780E] ; Find all the MS780-E's
            ;4287      ; ... on the SBI
U 1051, 0000,003C,0180,0800,0000,138B  ;4288  J/TEST.30.EXIT  ; No MS780-E's found
U 1052, 0000,003C,0180,0800,0000,1060  ;4289  nop   ;
            ;4290 =
            ;4291 =00
            ;4292 RESTART.TEST.30:  ; Entry point for second controller
            ;4293      ; ... and or next MS780-E
U 1060, 0000,003D,0180,0800,0000,105C  ;4294  CALL[START.TEST] ; Configure Controllers
U 1061, 0000,003C,0180,0800,0000,138B  ;4295  J/TEST.30.EXIT  ; No more Controllers or MS780-E's
U 1062, 0000,003C,0180,0800,0000,1064  ;4296  nop
            ;4297 =
U 1064, 0000,003D,0180,0800,0000,11C6  ;4298 =00 CALL[64K.OR.256K] ; Find out what kind of memory arrays are
            ;4299      ; present (64K or 256K)
U 1065, 0000,003C,0180,0800,0000,12FF  ;4300  J/T.30.MIX.ERROR ; CASE 1:  Return here when there is a mixture
            ;4301      ; of arrays on the controller or there were
            ;4302      ; no arrays found.
U 1066, 0000,003C,0180,0800,0000,1300  ;4303  J/T.30.64K.CHIPS ; CASE 2:  Return here when there is 64k ram
            ;4304      ; array boards present (1 mega byte array)
U 1067, 0000,003C,0180,0800,0000,1304  ;4305  J/T.30.256K.CHIPS ; CASE 3:  Return here when there is 256k ram
            ;4306      ; array boards present (4 mega byte array)
```

```
            ;4307  .PAGE
            ;4308 ;
            ;4309 ;////////////////////////////////////////////////////////////////////////////
            ;4310 ;
            ;4311 ; Got an error from 64K.OR.256K subroutine.  Either there are no arrays present
            ;4312 ; or there are a mixture of 64K and 256K arrays.
            ;4313 ;
            ;4314 T.30.MIX.ERROR:
U 12FF, 0000,003D,0980,0800,0084,704C  ;4315  ERROR1[.2]  ; Call out the error
            ;4316 ;
            ;4317 ;////////////////////////////////////////////////////////////////////////////
            ;4318 ;
            ;4319 ; DATA LOGOUT:
            ;4320 ;
            ;4321 ; DATA: I/O BASE ADDRESS OF MS780-E/H CURRENTLY UNDER TEST
            ;4322 ;  CNFG REGISTER C OR D FOR CONTROLLER UNDER TEST
            ;4323 ;
            ;4324 ;
            ;4325 ;////////////////////////////////////////////////////////////////////////////
            ;4326 ;
            ;4327 T.30.64K.CHIPS:
U 1300, 0003,003C,0180,09C8,0000,1301  ;4328  RC[09]_0  ; Clear flag to indicate that we have 64K chips
            ;4329      ; on the array (BIT 0)
U 1301, 0000,003C,2180,0800,0084,7302  ;4330  SC_K[.14]  ; Set-up to mask bit 20
U 1302, 0803,001C,0180,0800,0000,1303  ;4331  D_NOT.MASK  ; Assert bit 20 in D register to get x100000
            ;4332  RC[07]_D,        ; Move x100000 to RC[07] to set-up board size
U 1303, 0001,003C,0180,09B8,0000,1068  ;4333   J/T30.FIND.TR ; ... Go get TR of this memory
            ;4334 T.30.256K.CHIPS:
U 1304, 0018,0038,0580,09C8,0000,1305  ;4335  RC[09]_K[.1]  ; Set flag to indicate that 256K chips were
            ;4336      ; found on the array. (Bit 0)
U 1305, 0000,003C,2180,0800,0084,7306  ;4337  SC_K[.14]  ; Set up to mask bit 22
U 1306, 0000,003C,0980,0800,0084,9307  ;4338  SC_SC+K[.2]  ; ....
U 1307, 0803,001C,0180,0800,0000,1308  ;4339  D_NOT.MASK  ; Set bit 22 in the D register to get 400000
U 1308, 0001,003C,0180,09B8,0000,1068  ;4340  RC[07]_D          ; Move x40000 to RC[07] tp set up board size
            ;4341 =00
            ;4342 T30.FIND.TR:
U 1068, 0000,003D,0180,0800,0000,1246  ;4343  CALL[ODD.EVEN.TR] ; Get the TR of this memory
            ;4344  RC[0A]_K[.8],  ; RETURN1: Set flag for Evevn TR
U 1069, 0018,0038,0180,09D0,0000,1142  ;4345   J/T30.MAX.ADDR ; Go get the max address
            ;4346
U 106A, 0003,003C,0180,09D0,0000,1142  ;4347  RC[0A]_0  ; RETURN2: Set flag for ODD TR
            ;4348 =
            ;4349 =0
            ;4350 T30.MAX.ADDR:
U 1142, 0000,003D,0180,0800,0000,1230  ;4351  CALL[GET.MEMORY.SIZE] ; Get the max address + 1
U 1143, 0001,003C,0180,09B0,0000,1309  ;4352  RC[06]_D  ; ... save it in RC[06]
U 1309, 0000,003C,0580,0800,0084,730A  ;4353  SC_K[.1]  ; Set up to clear bit 1
U 130A, 0800,0038,0180,0800,0000,130B  ;4354  D_MASK   ; ....
U 130B, 0010,0038,01C0,0948,0000,130C  ;4355  Q_RC[09]  ; ....
U 130C, 001D,0034,0180,09C8,0000,1144  ;4356  RC[09]_D.AND.Q  ; Clear flag to indicate Operation
            ;4357      ; ... in Non-Interleaved mode (Bit 1)
            ;4358 ;
            ;4359 ;////////////////////////////////////////////////////////////////////////////
            ;4360 ;
            ;4361 ; SUBTEST 1 BOARD SELECT ADDRESSING (Non-Intrlvd)
```

```
            ;4362 ;
            ;4363 ;////////////////////////////////////////////////////////////////////////////////
            ;4364 ;
U 1144. 0000.003D.0180.0800.0000.1158  ;4365 =0 CALL[TEST.30.BOARD] ; Run the Board addressing test
            ;4366     ; ... in Non-Interleaved Mode
U 1145. 0000.003C.0180.0800.0000.130D  ;4367  nop   ;
            ;4368 ;////////////////////////////////////////////////////////////////////////////////
            ;4369 ;
            ;4370 ; End SUBTEST 1
            ;4371 ;
            ;4372 ;////////////////////////////////////////////////////////////////////////////////
            ;4373 ;
            ;4374 ; START SUBTEST 2 BOARD SELECT ADDRESSING (Ext-Intrlvd)
            ;4375 ;
            ;4376 ;////////////////////////////////////////////////////////////////////////////////
            ;4377 ;
U 130D. 0818.0038.0980.0800.0000.130E  ;4378  D_K[.2]   ; ...
U 130E. 0010.0038.01C0.0948.0000.130F  ;4379  Q_RC[09]  ; Set up to mask bit 1
U 130F. 001D.0030.0180.09C8.0000.1310  ;4380  RC[09]_D.OR.Q          ; Set flag indicating Externally
            ;4381     ; ...Interleaved mode
U 1310. 0000.003C.0980.0800.0084.7311  ;4382  SC_K[.2]  ; Set mask to find out what controller
            ;4383     ; ....to use UPPER/LOWER
U 1311. 0810.0038.0180.0968.0000.1312  ;4384  D_RC[0D]  ; Load D with RC[0D]
            ;4385  ALU_D.ANDNOT.MASK, ; Check bit <2> of RC[0D] for a x1
U 1312. 0001.0024.0180.0800.0010.1313  ;4386   CLK.UBCC ; Check Condition codes
U 1313. 0000.013C.0180.0800.0000.1146  ;4387  Z?   ; Check conditions codes
U 1146. 0000.003C.0180.0800.0000.1070  ;4388 =0 J/TST30.S2.UPPER ; Go select UPPER Controller
U 1147. 0000.003C.0180.0800.0000.1074  ;4389  J/TST30.S2.LOWER ; Go select LOWER Controller
            ;4390 ;
            ;4391 ; Enter here when selecting the UPPER controller.
            ;4392 ;
            ;4393 =00
            ;4394 TST30.S2.UPPER:
            ;4395  D_K[.3],  ; Set code for UPPER controller
U 1070. 0818.0039.0D80.0800.0000.11BC  ;4396  CALL[SELECT.CNTRLR] ; Select Upper Externally Interleaved
U 1071. 0000.003C.0180.0800.0000.1072  ;4397  nop   ; ...
U 1072. 0018.0038.0580.0AE0.0000.1148  ;4398  R[0C]_K[.1]  ; Set flag to indicate that we are
            ;4399 =     ; ... testing the Upper Controller
U 1148. 0000.003D.0180.0800.0000.1158  ;4400 =0 CALL[TEST.30.BOARD] ; Run the Board Addressing test in
            ;4401     ; ... Externally Interleaved Mode
U 1149. 0000.003C.0180.0800.0000.1314  ;4402  J/TST30.SUB.3  ; When RETURNED Go start Subtest 3
            ;4403 ;
            ;4404 ; Enter here when selecting the LOWER controller.
            ;4405 ;
            ;4406 =00
            ;4407 TST30.S2.LOWER:
            ;4408  D_K[.1],  ; Set Code for LOWER controller
U 1074. 0818.0039.0580.0800.0000.11BC  ;4409  CALL[SELECT.CNTRLR] ; Select Lower Externally Interleaved
U 1075. 0000.003C.0180.0800.0000.1076  ;4410  nop   ; ...
U 1076. 0003.003C.0180.0AE0.0000.114A  ;4411  R[0C]_0   ; Clear flag to indicate that we are
            ;4412 =     ; ... testing the Lower Controller
U 114A. 0000.003D.0180.0800.0000.1158  ;4413 =0 CALL[TEST.30.BOARD] ; Run the Board Addressing test in
            ;4414     ; ... Externally Interleaved Mode
U 114B. 0000.003C.0180.0800.0000.1314  ;4415  NOP
            ;4416 ;////////////////////////////////////////////////////////////////////////////////
```

```
            ;4417 ;
            ;4418 ; END SUBTEST 2
            ;4419 ;
            ;4420 ;////////////////////////////////////////////////////////////////////////////////
            ;4421 ;
            ;4422 ; SUBTEST 3  BANK SELECT ADDRESSING (Non-Intrlvd)
            ;4423 ;
            ;4424 ;////////////////////////////////////////////////////////////////////////////////
            ;4425 ;
            ;4426 TST30.SUB.3:
U 1314. 0000.003C.0180.0A60.0010.1315  ;4427  ALU_R[0C],CLK.UBCC ; See what controller we are testing
U 1315. 0000.013C.0180.0800.0000.114C  ;4428  Z?    ; Check conditions codes
U 114C. 0000.003C.0180.0800.0000.1078  ;4429 =0 J/TST30.S3.UPPER ; Go select UPPER Controller
U 114D. 0000.003C.0180.0800.0000.1080  ;4430  J/TST30.S3.LOWER ; Go select LOWER Controller
            ;4431 ;
            ;4432 ; Enter here when selecting the UPPER controller
            ;4433 ;
            ;4434 =00
            ;4435 TST30.S3.UPPER:
            ;4436  D_k[.2],  ; Set code for controller
U 1078. 0818.0039.0980.0800.0000.11BC  ;4437  CALL[SELECT.CNTRLR] ; Select UPPER Controller
            ;4438      ; ...Non-Interleaved Mode
U 1079. 0000.003C.0180.0800.0000.107A  ;4439  nop    ; ...
U 107A. 0000.003C.0580.0800.0084.7316  ;4440  SC_K[.1]   ; Set up to clear bit 1
U 1316. 0800.0038.0180.0800.0000.1317  ;4441 = D_MASK    ; ...
U 1317. 0010.0038.01C0.0948.0000.1318  ;4442  Q_RC[09]   ; ....
U 1318. 001D.0034.0180.09C8.0000.114E  ;4443  RC[09]_D.AND.Q  ; Clear flag to indicate Operation
            ;4444      ; ... in Non-Interleaved mode (Bit 1)
U 114E. 0000.003D.0180.0800.0000.117C  ;4445 =0 CALL[TEST.30.BANK] ; Run the Bank Test
U 114F. 0000.003C.0180.0800.0000.1151  ;4446  J/TST30.SUB.4  ; When RETURNED go start Subtest 4
            ;4447 ;
            ;4448 ; Enter here when selecting the LOWER controller
            ;4449 ;
            ;4450 =00
            ;4451 TST30.S3.LOWER:
            ;4452  D_0,    ; Set code for controller
U 1080. 0F00.003D.0180.0800.0000.11BC  ;4453  CALL[SELECT.CNTRLR] ; Select LOWER Controller
            ;4454      ; ...Non-Interleaved Mode
U 1081. 0000.003C.0180.0800.0000.1082  ;4455  nop    ; ....
U 1082. 0000.003C.0580.0800.0084.7319  ;4456  SC_K[.1]   ; Set up to clear bit 1
U 1319. 0800.0038.0180.0800.0000.131A  ;4457 = D_MASK    ; ...
U 131A. 0010.0038.01C0.0948.0000.131B  ;4458  Q_RC[09]   ; ....
U 131B. 001D.0034.0180.09C8.0000.1150  ;4459  RC[09]_D.AND.Q  ; Clear flag to indicate Operation
            ;4460      ; ... in Non-Interleaved mode (Bit 1)
U 1150. 0000.003D.0180.0800.0000.117C  ;4461 =0 CALL[TEST.30.BANK] ; Run the Bank Test
            ;4462 ; NOP
            ;4463 ;
            ;4464 ;////////////////////////////////////////////////////////////////////////////////
            ;4465 ;
            ;4466 ; END SUBTEST 3
            ;4467 ;
            ;4468 ;////////////////////////////////////////////////////////////////////////////////
            ;4469 ;
            ;4470 ; SUBTEST 4  BANK SELECT ADDRESSING (Ext-Intrlvd)
            ;4471 ;
```

```
            ;4472 ;////////////////////////////////////////////////////////////////////////////
            ;4473 ;
            ;4474 TST30.SUB.4:
U 1151, 0000,003C,0180,0A60,0010,131C  ;4475  ALU_R[0C],CLK.UBCC ; See what controller we are testing
U 131C, 0000,013C,0180,0800,0000,1152  ;4476  Z?    ; Check conditions codes
U 1152, 0000,003C,0180,0800,0000,1084  ;4477 =0 J/TST30.S4.UPPER ; Go select UPPER Controller
U 1153, 0000,003C,0180,0800,0000,1088  ;4478  J/TST30.S4.LOWER ; Go select LOWER Controller
            ;4479 ;
            ;4480 ; Enter here when selecting the UPPER controller
            ;4481 ;
U 131D, 0000,003C,0180,0800,0000,1084  ;4482  NOP
            ;4483 =00
            ;4484 TST30.S4.UPPER:
            ;4485  D_K[.3],  ; Set code for controller
U 1084, 0818,0039,0D80,0800,0000,11BC  ;4486  CALL[SELECT.CNTRLR] ; Select UPPER Controller
U 1085, 0000,003C,0180,0800,0000,1086  ;4487  nop   ; ...
U 1086, 0818,0038,0980,0800,0000,131E  ;4488  D_K[.2]   ;
            ;4489 =
U 131E, 0010,0038,01C0,0948,0000,131F  ;4490  Q_RC[09]  ; Set up to mask bit 1
U 131F, 001D,0030,0180,09C8,0000,1154  ;4491  RC[09]_D.OR.Q          ; Set flag indicating Externally
            ;4492     ; ...Interleaved mode
U 1154, 0000,003D,0180,0800,0000,117C  ;4493 =0 CALL[TEST.30.BANK] ; Run the Bank Addressing Test
U 1155, 0000,003C,0180,0800,0000,1060  ;4494  J/RESTART.TEST.30 ; When RETURNED Go see if there are any more
            ;4495     ; ....MS780-E's
            ;4496 ;
            ;4497 ; Enter here when selecting the LOWER controller
            ;4498 ;
            ;4499 =00
            ;4500 TST30.S4.LOWER:
            ;4501  D_K[.1],  ; Set code for controller
U 1088, 0818,0039,0580,0800,0000,11BC  ;4502  CALL[SELECT.CNTRLR] ; Select LOWER Controller
U 1089, 0000,003C,0180,0800,0000,108A  ;4503  nop   ; ....
U 108A, 0818,0038,0980,0800,0000,1320  ;4504  D_K[.2]   ; ...
            ;4505 =
U 1320, 0010,0038,01C0,0948,0000,1321  ;4506  Q_RC[09]  ; Set up to mask bit 1
U 1321, 001D,0030,0180,09C8,0000,1156  ;4507  RC[09]_D.OR.Q          ; Set flag indicating Externally
            ;4508     ; ...Interleaved mode
U 1156, 0000,003D,0180,0800,0000,117C  ;4509 =0 CALL[TEST.30.BANK] ; Run the Bank Addressing Test
U 1157, 0000,003C,0180,0800,0000,1060  ;4510  J/RESTART.TEST.30 ; When RETURNED Go see if there are any more
            ;4511     ; ....MS780-E's
            ;4512 ;
            ;4513 ; Enter here for the main body of the Board Addressing Test
            ;4514 ;
            ;4515 =0
            ;4516 TEST.30.BOARD:
U 1158, 0000,003D,0180,0800,0000,11BA  ;4517  SUBTEST   ; Increment the Subtest number
U 1159, 0003,003C,0180,0800,1408,715A  ;4518  STATE_0(A)  ; Clear the error counter
U 115A, 0000,003D,0180,0800,0000,10CC  ;4519 =0 CALL[SBI.INIT]  ; Clear SBI
U 115B, 0003,003C,0180,09C0,0000,1322  ;4520  RC[08]_0  ; Clear the board pointer
U 1322, 0003,003C,0180,09A0,0000,1323  ;4521  RC[04]_0  ; Clear the pattern counter
U 1323, 0003,003C,0180,0980,0000,115C  ;4522  RC[1]_0   ; Pattern of zeroes for initialization
            ;4523 =0
            ;4524 TEST.30.LOOP1:
U 115C, 0000,003D,0180,0800,0000,124C  ;4525  CALL[GEN.BOARD.ADD] ; Generate the board address
U 115D, 0001,003C,0180,0800,0200,1324  ;4526  VA_D   ; Point to location 0 of the board under test
```

```
U 1324, 0001,003C,0180,09D8,0000,1325  ;4527  RC[0B]_D  ; Save address for error logout
U 1325, 0810,0038,0180,0908,0000,1326  ;4528  D_RC[01]  ; Get encoded patter of 0's to D register
U 1326, 0000,003C,0180,A800,0000,10A0  ;4529  CACHE.P_D[LONG]  ; Write the pattern to the board address in VA
U 10A0, 0000,003D,0180,0800,0000,11E0  ;4530 =00 CHECK.INTERRUPT  ; Check for interrupts
U 10A1, 0000,003C,0180,0800,0000,1327  ;4531  J/TST30.NO.INT  ; Jump if no interrupts else
```

```
          ;4532  .PAGE
          ;4533 ;
          ;4534 ;////////////////////////////////////////////////////////////////////////////////
          ;4535 ;
U 10A2, 0000,003D,F580,0800,0084,704C  ;4536  ERROR1[.A]  ; Unexpected Interrupt occured when writting
          ;4537      ; ...the encoded pattern of 0's to selected
          ;4538      ; ...board
          ;4539 ;
          ;4540 ;////////////////////////////////////////////////////////////////////////////////
          ;4541 ;
          ;4542 ; ERROR LOGOUT:
          ;4543 ;
          ;4544 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
          ;4545 ;  CNFG C/D ADDRESS OF CURRENTLY TESTED MS780E
          ;4546 ;  NOT USED
          ;4547 ;  REFERENCED ADDRESS THAT CAUSED INTERRUPT
          ;4548 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
          ;4549 ;  INTERLEAVED MODE BITS OF CNFG REG A
          ;4550 ;  INTERRUPT VECTOR REGISTER
          ;4551 ;  SBI.ERROR REGISTER
          ;4552 ;  SBI.FAULT REGISTER
          ;4553 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
          ;4554 ;
          ;4555 ;////////////////////////////////////////////////////////////////////////////////
          ;4556 ;
          ;4557 =
          ;4558 TST30.NO.INT:
U 1327, 0010,0038,01C0,0938,0000,1328  ;4559  Q_RC[07]  ; Move Displacement to Q
U 1328, 0810,0038,0180,0940,0000,1329  ;4560  D_RC[08]  ; Move board pointer to D
U 1329, 001D,0014,0180,09C0,0000,132A  ;4561  ALU_D+Q,RC[08]_ALU ; Point to next board
U 132A, 0810,0038,0180,0940,0000,132B  ;4562  D_RC[08]  ; Move board pointer to D register
          ;4563  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 132B, 0011,0020,0180,0930,0010,132C  ;4564   CLK.UBCC ; ... if they are equal we are finished
U 132C, 0000,013C,0180,0800,0000,115E  ;4565  Z?   ; Are they equal ?
U 115E, 0000,003C,0180,0800,0000,115C  ;4566 =0 J/TEST.30.LOOP1  ; No continue writing encoded pattern of 0's
          ;4567      ; ... Yes continue with next part of test
U 115F, 0003,003C,0180,09C0,0000,132D  ;4568  RC[08]_0  ; Clear board counter
U 132D, 0018,0038,0580,09A0,0000,1160  ;4569  RC[04]_K[.1]  ; Initialize the pattern counter
          ;4570 =0
          ;4571 TEST.30.LOOP2:
U 1160, 0000,003D,0180,0800,0000,124C  ;4572  CALL[GEN.BOARD.ADD] ; Generate the board address
U 1161, 0001,003C,0180,0800,0200,132E  ;4573  VA_D   ; Point to location 0 of the board under test
U 132E, 0001,003C,0180,09D8,0000,132F  ;4574  RC[0B]_D  ; Save for error information
U 132F, 0010,0038,01C0,0920,0000,1330  ;4575  Q_RC[04]  ; Save the pattern counter
U 1330, 0001,203C,0180,0AE8,0000,1331  ;4576  R[0D]_Q        ; ...in R[0D]
U 1331, 0003,003C,0180,09A0,0000,1332  ;4577  RC[04]_0  ; Set pattern counter to 0
U 1332, 0003,003C,0180,0AF8,0000,1162  ;4578  R[0F]_0   ; Clear Utrap flag
U 1162, 0000,003D,8980,0800,0084,71ED  ;4579 =0 SET.TRAP.MASK[.D] ; Enable Time-Out Utraps
U 1163, 0000,003C,0180,C800,0000,1333  ;4580  D[LONG]_CACHE.P  ; Read the location written
U 1333, 0001,003C,0180,0980,0000,1334  ;4581  RC[0]_D   ; Save in RC[0] for decode
          ;4582  ALU_R[0F],  ; Check to see if a Time-Out Utrap occured
U 1334, 0000,003C,0180,0A78,0010,1335  ;4583   CLK.UBCC ; Clock Condition Codes
U 1335, 0000,013C,0180,0800,0000,1164  ;4584  Z?   ; Is it Zero?
U 1164, 0000,003C,0180,0800,0000,10A4  ;4585 =0 J/TST30.L2.NOTIMO ; Jump if no Time-Outs occured else...
```

```
            ;4586  .PAGE
            ;4587 ;
            ;4588 ;////////////////////////////////////////////////////////////////////////////////
            ;4589 ;
U 1165, 0000,003D,D580,0800,0084,704C  ;4590  ERROR1[.6]  ; Time-Out on READ occured
            ;4591 ;
            ;4592 ;
            ;4593 ;////////////////////////////////////////////////////////////////////////////////
            ;4594 ;
            ;4595 ; ERROR LOGOUT:
            ;4596 ;
            ;4597 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4598 ;  CNFG C/D ADDRESS OF CURRENTLY TESTED MS780E
            ;4599 ;  NOT USED
            ;4600 ;  REFERENCED ADDRESS THAT CAUSED TIME=OUT
            ;4601 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
            ;4602 ;  INTERLEAVED MODE BITS OF CNFG REG A
            ;4603 ;
            ;4604 ;////////////////////////////////////////////////////////////////////////////////
            ;4605 ;
            ;4606 =00
            ;4607 TST30.L2.NOTIMO:
U 10A4, 0000,003D,0180,0800,0000,11E0  ;4608  CHECK.INTERRUPT  ; Go check for Unexpected Interrupts
U 10A5, 0000,003C,0180,0800,0000,10A8  ;4609  J/TST30.L2.NOINT ; Jump if no interrupts else ...
            ;4610 ;
            ;4611 ;////////////////////////////////////////////////////////////////////////////////
            ;4612 ;
U 10A6, 0000,003D,F580,0800,0084,704C  ;4613  ERROR1[.A]  ; Unexpected Interrupt
            ;4614 ;
            ;4615 ;////////////////////////////////////////////////////////////////////////////////
            ;4616 ;
            ;4617 ; ERROR LOGOUT:
            ;4618 ;
            ;4619 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4620 ;  NOT USED
            ;4621 ;  NOT USED
            ;4622 ;  REFERENCED ADDRESS THAT CAUSED INTERRUPT
            ;4623 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
            ;4624 ;  INTERLEAVED MODE BITS OF CNFG REG A
            ;4625 ;  INTERRUPT VECTOR REGISTER
            ;4626 ;  SBI.ERROR REGISTER
            ;4627 ;  SBI.FAULT REGISTER
            ;4628 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;4629 ;
            ;4630 ;////////////////////////////////////////////////////////////////////////////////
            ;4631 =
            ;4632 =00
            ;4633 TST30.L2.NOINT:
U 10A8, 0000,003D,0180,0800,0000,1283  ;4634  CALL[DECODE]  ; Are the 0's previously written still there ?
U 10A9, 0000,003C,0180,0800,0000,1166  ;4635  J/TST30.MAX.ERR1 ; Maximum amount of errors occured call error
U 10AA, 0000,003C,0180,0800,0000,10AB  ;4636  nop   ; An acceptable amount of errors occured cont..
U 10AB, 0000,003C,0180,0800,0000,1169  ;4637  J/TST30.BOD.CONT1 ; No errors continue test
            ;4638 =0
            ;4639 TST30.MAX.ERR1:
U 1166, 0000,003D,0180,0800,0000,1237  ;4640  CALL[ERDAT]  ; Get logout data
```

```
U 1167, 0000,003C,0180,0800,0000,1168  ;4641  nop
```

```
                ;4642  .PAGE
                ;4643 ;
                ;4644 ;////////////////////////////////////////////////////////////////////////////////
                ;4645 ;
U 1168, 0000,003D,6180,0800,0084,704C  ;4646 =0 ERROR1[.F]  ; Log the error
                ;4647 ;
                ;4648 ;////////////////////////////////////////////////////////////////////////////////
                ;4649 ;
                ;4650 ;  I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
                ;4651 ;  NOT USED
                ;4652 ;  Received Memory Data
                ;4653 ;  Expected Memory Data
                ;4654 ;  Received Memory Data
                ;4655 ;  Expected Memory Data
                ;4656 ;  Received Memory Data
                ;4657 ;  Expected Memory Data
                ;4658 ;  Received Memory Data
                ;4659 ;  Expected Memory Data
                ;4660 ;  Received Memory Data
                ;4661 ;  Expected Memory Data
                ;4662 ;  Received Memory Data
                ;4663 ;  Expected Memory Data
                ;4664 ;  FFFFFFFF
                ;4665 ;
                ;4666 ; NOTE:  The combination "Received and Expected Memory Data" will be
                ;4667 ;        terminated by a xFFFFFFFF.  If there are more errors then can
                ;4668 ;        be output on the error then the last longword wrote will be the
                ;4669 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
                ;4670 ;
                ;4671 ;////////////////////////////////////////////////////////////////////////////////
                ;4672 ;
                ;4673 TST30.BOD.CONT1:
U 1169, 0800,003C,0180,0A68,0000,1336  ;4674  D_R[0D]   ; Moved saved pattern counter to the D register
U 1336, 0001,003C,0180,09A0,0000,116A  ;4675  RC[04]_D  ; Restore the pattern count in RC[04]
U 116A, 0000,003D,0180,0800,0000,125F  ;4676 =0 CALL[ENCODE]  ; Encode the pattern
U 116B, 0010,0038,0180,0958,0200,1337  ;4677  VA_RC[0B]  ; Point to location under test
U 1337, 0810,0038,0180,0908,0000,1338  ;4678  D_RC[01]  ; Move the encoded pattern to the D register
U 1338, 0000,003C,0180,A800,0000,1339  ;4679  CACHE.P_D[LONG]  ; Write the Encoded pattern
U 1339, 0810,0038,0180,0920,0000,133A  ;4680  D_RC[04]  ; Move pattern count to the D register
U 133A, 0019,0014,0580,09A0,0000,133B  ;4681  RC[04]_D+K[.1]  ; Point to the next pattern
                ;4682 ;
                ;4683 ; Set up to point to next board
                ;4684 ;
U 133B, 0010,0038,01C0,0938,0000,133C  ;4685  Q_RC[07]  ; Move Displacement to Q
U 133C, 0810,0038,0180,0940,0000,133D  ;4686  D_RC[08]  ; Move board pointer to D
U 133D, 001D,0014,0180,09C0,0000,133E  ;4687  ALU_D+Q,RC[08]_ALU ; Point to next board
U 133E, 0810,0038,0180,0940,0000,133F  ;4688  D_RC[08]  ; Move board pointer to D register
                ;4689  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 133F, 0011,0020,0180,0930,0010,1340  ;4690   CLK.UBCC ; ... if they are equal we are finished
U 1340, 0000,013C,0180,0800,0000,116C  ;4691  Z?   ; Are they equal ?
U 116C, 0000,003C,0180,0800,0000,1160  ;4692 =0 J/TEST.30.LOOP2  ; No continue writting encoded pattern of 0's
                ;4693      ; ... Yes continue with next part of test
U 116D, 0003,003C,0180,09C0,0000,1341  ;4694  RC[08]_0  ; Clear board counter
U 1341, 0018,0038,0580,09A0,0000,116E  ;4695  RC[04]_K[.1]  ; Initialize the pattern counter
                ;4696 =0
```

```
            ;4697 TEST.30.LOOP3:
U 116E. 0000.003D.0180.0800.0000.124C  ;4698  CALL[GEN.BOARD.ADD] ; Generate the board address
U 116F. 0001.003C.0180.0800.0200.1342  ;4699  VA_D    ; Point to location 0 of the board under test
U 1342. 0000.003C.0180.C800.0000.1343  ;4700  D[LONG]_CACHE.P  ; Read the location written
U 1343. 0001.003C.0180.0980.0000.10B0  ;4701  RC[0]_D    ; Save for decode
U 10B0. 0000.003D.0180.0800.0000.1283  ;4702 =00 CALL[DECODE]  ; See if the correct data was written
U 10B1. 0000.003C.0180.0800.0000.1170  ;4703  J/TST30.MAX.ERR2 ; Maximum amount of errors occured call error
U 10B2. 0000.003C.0180.0800.0000.10B3  ;4704  nop    ; An acceptable amount of errors occured cont..
U 10B3. 0000.003C.0180.0800.0000.1173  ;4705  J/TST30.BOD.CONT2 ; No errors continue test
            ;4706 =0
            ;4707 TST30.MAX.ERR2:
U 1170. 0000.003D.0180.0800.0000.1237  ;4708  CALL[ERDAT]  ; Get logout data
U 1171. 0000.003C.0180.0800.0000.1172  ;4709  nop
```

```
            :4710  .PAGE
            :4711 ;
            :4712 ;////////////////////////////////////////////////////////////////////////////////
            :4713 ;
U 1172, 0000.003D.6180.0800.0084.704C  :4714 =0 ERROR1[.F]  ; Log the error
            :4715 ;
            :4716 ;////////////////////////////////////////////////////////////////////////////////
            :4717 ;
            :4718 ;  I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            :4719 ;  Configuration Register C or D ( C = Lower & D = Upper)
            :4720 ;  Received Memory Data
            :4721 ;  Expected Memory Data
            :4722 ;  Received Memory Data
            :4723 ;  Expected Memory Data
            :4724 ;  Received Memory Data
            :4725 ;  Expected Memory Data
            :4726 ;  Received Memory Data
            :4727 ;  Expected Memory Data
            :4728 ;  Received Memory Data
            :4729 ;  Expected Memory Data
            :4730 ;  Received Memory Data
            :4731 ;  Expected Memory Data
            :4732 ;  FFFFFFFF
            :4733 ;
            :4734 ; NOTE:  The combination "Received and Expected Memory Data" will be
            :4735 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            :4736 ;        be output on the error then the last longword wrote will be the
            :4737 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            :4738 ;
            :4739 ;////////////////////////////////////////////////////////////////////////////////
            :4740 ;
            :4741 TST30.BOD.CONT2:
U 1173, 0810.0038.0180.0920.0000.1344  :4742  D_RC[04]          ; Move pattern count to the D register
U 1344, 0019.0014.0580.09A0.0000.1345  :4743  RC[04]_D+K[.1]  ; Point to the next pattern
            :4744 ;
            :4745 ; Set up to point to next board
            :4746 ;
U 1345, 0010.0038.01C0.0938.0000.1346  :4747  Q_RC[07]  ; Move Displacement to Q
U 1346, 0810.0038.0180.0940.0000.1347  :4748  D_RC[08]  ; Move board pointer to D
U 1347, 001D.0014.0180.09C0.0000.1348  :4749  ALU_D+Q,RC[08]_ALU ; Point to next board
U 1348, 0810.0038.0180.0940.0000.1349  :4750  D_RC[08]  ; Move board pointer to D register
            :4751  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 1349, 0011.0020.0180.0930.0010.134A  :4752   CLK.UBCC ; ... if they are equal we are finished
U 134A, 0000.013C.0180.0800.0000.1174  :4753  Z?   ; Are they equal ?
U 1174, 0000.003C.0180.0800.0000.116E  :4754 =0 J/TEST.30.LOOP3  ; No continue reading encoded pattern of 0's
            :4755     ; ... Yes continue with next part of test
            :4756  ;
            :4757  ; This is the last section of this test and here we check the value
            :4758  ; of state.  IF state is not equal to zero then get data for logout
            :4759  ; and call error1, Else return to the caller.
            :4760  ;
U 1175, 0000.003C.0180.0800.1480.134B  :4761  SC_STATE  ; Move STATE to the SC
U 134B, 0018.0038.1DC0.0800.0000.134C  :4762  Q_SC   ; Put SC (STATE) in Q
U 134C, 0001.203C.0180.0800.0010.134D  :4763  ALU_Q,CLK.UBCC  ; Move Q (State) into the ALU for check
U 134D, 0000.013C.0180.0800.0000.1176  :4764  Z?   ; Is state 0 ?
```

```
U 1176, 0000,003C,0180,0800,0000,1178  ;4765 =0 J/T.30.STATE.NE.ZERO ; No, get the data for logout and call an error
U 1177, 0000,003E,0180,0800,0000,0001  ;4766  RETURN1   ; Yes, return to the caller
            ;4767  ;
            ;4768  ; Come here after checking the value of state and finding that it is
            ;4769  ; non zero.  Get the data for the logout and call an error.
            ;4770  ;
            ;4771 =0
            ;4772 T.30.STATE.NE.ZERO:
U 1178, 0000,003D,0180,0800,0000,1237  ;4773  CALL[ERDAT]  ; Get the data for logout
U 1179, 0000,003C,0180,0800,0000,117A  ;4774  NOP
```

```
            ;4775  .PAGE
            ;4776 ;
            ;4777 ;////////////////////////////////////////////////////////////////////////////////
            ;4778 ;
            ;4779 ; This will call an error from state no equal to zero.
            ;4780 ;
U 117A, 0000.003D.6180.0800.0084.704C  ;4781 =0 ERROR1[.F]
U 117B, 0000.003E.0180.0800.0000.0001  ;4782  RETURN1
            ;4783 ;
            ;4784 ;////////////////////////////////////////////////////////////////////////////////
            ;4785 ;
            ;4786 ; ERROR LOGOUT:
            ;4787 ;
            ;4788 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4789 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4790 ;  Received Memory Data
            ;4791 ;  Expected Memory Data
            ;4792 ;  Received Memory Data
            ;4793 ;  Expected Memory Data
            ;4794 ;  Received Memory Data
            ;4795 ;  Expected Memory Data
            ;4796 ;  Received Memory Data
            ;4797 ;  Expected Memory Data
            ;4798 ;  Received Memory Data
            ;4799 ;  Expected Memory Data
            ;4800 ;  Received Memory Data
            ;4801 ;  Expected Memory Data
            ;4802 ;  FFFFFFFF
            ;4803 ;
            ;4804 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4805 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4806 ;        be output on the error then the last longword wrote will be the
            ;4807 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4808 ;
            ;4809 ;////////////////////////////////////////////////////////////////////////////////
            ;4810 ;
            ;4811 =0
            ;4812 TEST.30.BANK:
U 117C, 0000.003D.0180.0800.0000.11BA  ;4813  SUBTEST   ; Increment the Subtest number
U 117D, 0003.003C.0180.0800.1408.717E  ;4814  STATE_0(A)  ; Clear the error counter
U 117E, 0000.003D.0180.0800.0000.10CC  ;4815 =0 CALL[SBI.INIT]  ; Clear the SBI
U 117F, 0003.003C.0180.09C0.0000.134E  ;4816  RC[08]_0  ; Clear the board pointer
U 134E, 0003.003C.0180.09A0.0000.134F  ;4817  RC[04]_0  ; Clear the pattern counter
            ;4818 TST30.BANK.LOOP2:
U 134F, 0003.003C.0180.09D8.0000.1180  ;4819  RC[0B]_0  ; Clear the bank counter
            ;4820 =0
            ;4821 TST30.BANK.LOOP1:
U 1180, 0000.003D.0180.0800.0000.10EE  ;4822  CALL[GEN.BANK.ADD] ; Generate the Bank Address
U 1181, 0001.003C.0180.0800.0200.1350  ;4823  VA_D   ; Get address to the Va Register
U 1350, 0F00.003C.0180.0800.0000.1351  ;4824  D_0   ; Move 0's to the D Register
U 1351, 0000.003C.0180.A800.0000.10B4  ;4825  CACHE.P_D[LONG]  ; Write 0's to Bank location 0
U 10B4, 0000.003D.0180.0800.0000.11E0  ;4826 =00 CHECK.INTERRUPT  ; Any Unexpected Interrupts ?
U 10B5, 0000.003C.0180.0800.0000.1352  ;4827  J/TST30.BANK.NOINT1 ; Jump if No Interrupts else....
```

```
            ;4828  .PAGE
            ;4829 ;
            ;4830 ;////////////////////////////////////////////////////////////////////////////////
            ;4831 ;
U 10B6, 0000,003D,F580,0800,0084,704C  ;4832  ERROR1[.A]  ; Unexpected Interrupt on write to the array
            ;4833 ;
            ;4834 ;////////////////////////////////////////////////////////////////////////////////
            ;4835 ;
            ;4836 ; ERROR LOGOUT:
            ;4837 ;
            ;4838 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4839 ;  CNFG C/D ADDRESS OF CURRENTLY TESTED MS780E
            ;4840 ;  NOT USED
            ;4841 ;  BANK COUNTER
            ;4842 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
            ;4843 ;  INTERLEAVED MODE BITS OF CNFG REG A
            ;4844 ;  INTERRUPT VECTOR REGISTER
            ;4845 ;  SBI.ERROR REGISTER
            ;4846 ;  SBI.FAULT REGISTER
            ;4847 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;4848 ;
            ;4849 ;////////////////////////////////////////////////////////////////////////////////
            ;4850 ;
            ;4851 =
            ;4852 TST30.BANK.NOINT1:
U 1352, 0810,0038,0180,0958,0000,1353  ;4853  D_RC[0B]          ; Move Bank count to the D register
U 1353, 0019,0014,0580,09D8,0000,1354  ;4854  RC[0B]_D+K[.1]  ; Point to the next Bank
            ;4855 ;
            ;4856 ; See if we are done with all the Banks on this board
            ;4857 ;
U 1354, 0818,0038,1180,0800,0000,1355  ;4858  D_K[.4]   ; Move Max bank number to D register
            ;4859  ALU_D.XOR.RC[0B], ; Compare the bank counter to max bank
U 1355, 0011,0020,0180,0958,0010,1356  ;4860   CLK.UBCC ; ... if they are equal we are finished
U 1356, 0000,013C,0180,0800,0000,1182  ;4861  Z?    ; Are they equal ?
U 1182, 0000,003C,0180,0800,0000,1180  ;4862 =0 J/TST30.BANK.LOOP1 ; No continue writting encoded pattern of 0's
            ;4863     ; Yes continue with next part of test
            ;4864 ; Set up to point to next board
            ;4865 ;
U 1183, 0010,0038,01C0,0938,0000,1357  ;4866  Q_RC[07]  ; Move Displacement to Q
U 1357, 0810,0038,0180,0940,0000,1358  ;4867  D_RC[08]  ; Move board pointer to D
U 1358, 001D,0014,0180,09C0,0000,1359  ;4868  ALU_D+Q,RC[08]_ALU ; Point to next board
U 1359, 0810,0038,0180,0940,0000,135A  ;4869  D_RC[08]  ; Move board pointer to D register
            ;4870  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 135A, 0011,0020,0180,0930,0010,135B  ;4871   CLK.UBCC ; ... if they are equal we are finished
U 135B, 0000,013C,0180,0800,0000,1184  ;4872  Z?    ; Are they equal ?
U 1184, 0000,003C,0180,0800,0000,134F  ;4873 =0 J/TST30.BANK.LOOP2 ; No continue reading encoded pattern of 0's
            ;4874     ; ... Yes continue with next part of test
U 1185, 0003,003C,0180,09C0,0000,135C  ;4875  RC[08]_0  ; Clear the board counter
U 135C, 0018,0038,0580,09A0,0000,135D  ;4876  RC[04]_K[.1]  ; Initialize the Pattern counter
            ;4877 ;
            ;4878 TST30.BANK.LOOP4:
U 135D, 0003,003C,0180,09D8,0000,1186  ;4879  RC[0B]_0  ; Initialize the Bank counter
            ;4880 ;
            ;4881 =0
            ;4882 TST30.BANK.LOOP3:
```

```
U 1186, 0000.003D.0180.0800.0000.10EE  ;4883  CALL[GEN.BANK.ADD] ; Generate the Bank Address
U 1187, 0001.003C.0180.0800.0200.135E  ;4884  VA_D   ; Get address to the VA Register
U 135E, 0001.003C.0180.09E0.0000.135F  ;4885  RC[0C]_D  ; Save BANK Address
U 135F, 0003.003C.0180.0AF8.0000.1188  ;4886  R[0F]_0   ; Clear Utrap Flag
U 1188, 0000.003D.8980.0800.0084.71ED  ;4887 =0 SET.TRAP.MASK[.D] ; Enable Time-Outs Utraps
U 1189, 0000.003C.0180.C800.0000.1360  ;4888  D[LONG]_CACHE.P  ; Read the VA store Vector in
U 1360, 0001.003C.0180.0980.0000.1361  ;4889  RC[0]_D   ; ... RC[0] for decoding
          ;4890  ALU_R[0F],  ; Check to see if a Time-Out occured
U 1361, 0000.003C.0180.0A78.0010.1362  ;4891   CLK.UBCC ; Clock Condition codes
U 1362, 0000.013C.0180.0800.0000.118A  ;4892  Z?   ; Is it Zero ?
U 118A, 0000.003C.0180.0800.0000.10B8  ;4893 =0 J/TST30.BANK.NOTIMO ; Jump if no Time-Outs occured else...
```

```
            ;4894  .PAGE
            ;4895 ;
            ;4896 ;////////////////////////////////////////////////////////////////////////////
            ;4897 ;
U 118B, 0000,003D,D580,0800,0084,704C  ;4898  ERROR1[.6]  ; Time-Out on READ occured
            ;4899 ;
            ;4900 ;
            ;4901 ;////////////////////////////////////////////////////////////////////////////
            ;4902 ;
            ;4903 ; ERROR LOGOUT:
            ;4904 ;
            ;4905 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4906 ;  CNFG C/D ADDRESS OF CURRENTLY TESTED MS780E
            ;4907 ;  UPPER/LOWER FLAG (0=LOWER CNTRL, 1=UPPER CNTRL)
            ;4908 ;  BANK POINTER
            ;4909 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
            ;4910 ;  INTERLEAVED MODE BITS OF CNFG REG A
            ;4911 ;
            ;4912 ;////////////////////////////////////////////////////////////////////////////
            ;4913 ;
            ;4914 =00
            ;4915 TST30.BANK.NOTIMO:
U 10B8, 0000,003D,0180,0800,0000,11E0  ;4916  CHECK.INTERRUPT  ; Go check for Unexpected Interrupts
U 10B9, 0000,003C,0180,0800,0000,1363  ;4917  J/TST30.BANK.NOINT2 ; Jump if no interrupts else ...
            ;4918 ;
            ;4919 ;////////////////////////////////////////////////////////////////////////////
            ;4920 ;
U 10BA, 0000,003D,F580,0800,0084,704C  ;4921  ERROR1[.A]  ; Unexpected Interrupt
            ;4922 ;
            ;4923 ;////////////////////////////////////////////////////////////////////////////
            ;4924 ;
            ;4925 ; ERROR LOGOUT:
            ;4926 ;
            ;4927 ; DATA: I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4928 ;  CNFG C/D ADDRESS OF CURRENTLY TESTED MS780E
            ;4929 ;  UPPER/LOWER FLAG (1= UPPER CONTROLLER, 0= LOWER CONTROLLER)
            ;4930 ;  BANK POINTER
            ;4931 ;  CNTRL SELECT ACCORDING TO TR ODD/EVEN (0= EVEN, 8=ODD)
            ;4932 ;  INTERLEAVED MODE BITS OF CNFG REG A
            ;4933 ;  INTERRUPT VECTOR REGISTER
            ;4934 ;  SBI.ERROR REGISTER
            ;4935 ;  SBI.FAULT REGISTER
            ;4936 ;  INTERRUPT OCCURED FLAG (TRUE=1, FALSE=0)
            ;4937 ;
            ;4938 ;////////////////////////////////////////////////////////////////////////////
            ;4939 =
            ;4940 TST30.BANK.NOINT2:
U 1363, 0010,0038,01C0,0920,0000,1364  ;4941  Q_RC[04]  ; Save the pattern counter
U 1364, 0001,203C,0180,0AE8,0000,1365  ;4942  R[0D]_Q   ; ...in R[0D]
U 1365, 0003,003C,0180,09A0,0000,10C0  ;4943  RC[04]_0  ; Set pattern counter to ZERO
U 10C0, 0000,003D,0180,0800,0000,1283  ;4944 =00 CALL[DECODE]  ; Are the 0's previously written still there ?
U 10C1, 0000,003C,0180,0800,0000,118C  ;4945  J/TST30.BANK.MAX.ERR1 ; Maximum amount of errors occured call error
U 10C2, 0000,003C,0180,0800,0000,10C3  ;4946  nop   ; An acceptable amount of errors occured cont..
U 10C3, 0000,003C,0180,0800,0000,118F  ;4947  J/TST30.BANK.CONT1 ; No errors continue test
            ;4948 =0
```

```
            ;4949 TST30.BANK.MAX.ERR1:
U 118C, 0000.003D.0180.0800.0000.1237  ;4950  CALL[ERDAT]  ; Get logout data
U 118D, 0000.003C.0180.0800.0000.118E  ;4951  nop
```

```
            ;4952  .PAGE
            ;4953 ;
            ;4954 ;////////////////////////////////////////////////////////////////////////////////
            ;4955 ;
U 118E, 0000,003D,6180,0800,0084,704C  ;4956 =0 ERROR1[.F]  ; Log the error
            ;4957 ;
            ;4958 ;////////////////////////////////////////////////////////////////////////////////
            ;4959 ;
            ;4960 ;  I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;4961 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4962 ;  Received Memory Data
            ;4963 ;  Expected Memory Data
            ;4964 ;  Received Memory Data
            ;4965 ;  Expected Memory Data
            ;4966 ;  Received Memory Data
            ;4967 ;  Expected Memory Data
            ;4968 ;  Received Memory Data
            ;4969 ;  Expected Memory Data
            ;4970 ;  Received Memory Data
            ;4971 ;  Expected Memory Data
            ;4972 ;  Received Memory Data
            ;4973 ;  Expected Memory Data
            ;4974 ;  FFFFFFFF
            ;4975 ;
            ;4976 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4977 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4978 ;        be output on the error then the last longword wrote will be the
            ;4979 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4980 ;
            ;4981 ;////////////////////////////////////////////////////////////////////////////////
            ;4982 ;
            ;4983 TST30.BANK.CONT1:
U 118F, 0800,003C,0180,0A68,0000,1366  ;4984  D_R[0D]   ; Moved saved pattern counter to the D register
U 1366, 0001,003C,0180,09A0,0000,1190  ;4985  RC[04]_D  ; Restore the pattern count in RC[04]
U 1190, 0000,003D,0180,0800,0000,125F  ;4986 =0 CALL[ENCODE]  ; Encode the pattern
U 1191, 0010,0038,0180,0960,0200,1367  ;4987  VA_RC[0C]  ; Point to the BANK address under test
U 1367, 0810,0038,0180,0908,0000,1368  ;4988  D_RC[01]  ; Move the encoded pattern to the D register
U 1368, 0000,003C,0180,A800,0000,1369  ;4989  CACHE.P_D[LONG]  ; Write the Encoded pattern
U 1369, 0810,0038,0180,0920,0000,136A  ;4990  D_RC[04]  ; Move pattern count to the D register
U 136A, 0019,0014,0580,09A0,0000,136B  ;4991  RC[04]_D+K[.1]  ; Point to the next pattern
U 136B, 0810,0038,0180,0958,0000,136C  ;4992  D_RC[0B]        ; Move Bank count to the D register
U 136C, 0019,0014,0580,09D8,0000,136D  ;4993  RC[0B]_D+K[.1]  ; Point to the next Bank
            ;4994 ;
            ;4995 ; See if we are done with all the Banks on this board
            ;4996 ;
U 136D, 0818,0038,1180,0800,0000,136E  ;4997  D_K[.4]   ; Move Max bank number to D register
            ;4998  ALU_D.XOR.RC[0B], ; Compare the bank counter to max bank
U 136E, 0011,0020,0180,0958,0010,136F  ;4999   CLK.UBCC ; ... if they are equal we are finished
U 136F, 0000,013C,0180,0800,0000,1192  ;5000  Z?    ; Are they equal ?
U 1192, 0000,003C,0180,0800,0000,1186  ;5001 =0 J/TST30.BANK.LOOP3 ; No continue READING encoded pattern of 0's
            ;5002      ; Yes continue with next part of test
            ;5003 ;
            ;5004 ; Set up to point to next board
            ;5005 ;
U 1193, 0010,0038,01C0,0938,0000,1370  ;5006  Q_RC[07]  ; Move Displacement to Q
```

```
U 1370, 0810,0038,0180,0940,0000,1371  ;5007  D_RC[08]  ; Move board pointer to D
U 1371, 001D,0014,0180,09C0,0000,1372  ;5008  ALU_D+Q,RC[08]_ALU ; Point to next board
U 1372, 0810,0038,0180,0940,0000,1373  ;5009  D_RC[08]  ; Move board pointer to D register
          ;5010  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 1373, 0011,0020,0180,0930,0010,1374  ;5011   CLK.UBCC ; ... if they are equal we are finished
U 1374, 0000,013C,0180,0800,0000,1194  ;5012  Z?   ; Are they equal ?
U 1194, 0000,003C,0180,0800,0000,135D  ;5013 =0 J/TST30.BANK.LOOP4 ; No continue WRITING encoded pattern of 0's
          ;5014      ; ... Yes continue with next part of test
U 1195, 0000,003C,0180,0800,0000,1196  ;5015  NOP
U 1196, 0000,003D,0180,0800,0000,12FE  ;5016 =0 CALL[DELAY]  ; Clear Time-Out counter in Mic Mon
U 1197, 0003,003C,0180,09C0,0000,1375  ;5017  RC[08]_0  ; Clear board counter
U 1375, 0018,0038,0580,09A0,0000,1376  ;5018  RC[04]_K[.1]  ; Initialize the pattern counter
          ;5019 ;
          ;5020 TST30.BANK.LOOP6:
U 1376, 0003,003C,0180,09D8,0000,1198  ;5021  RC[0B]_0  ; Clear the Bank Counter
          ;5022 =0
          ;5023 TST30.BANK.LOOP5:
U 1198, 0000,003D,0180,0800,0000,10EE  ;5024  CALL[GEN.BANK.ADD] ; Generate the Bank Address
U 1199, 0001,003C,0180,0800,0200,1377  ;5025  VA_D   ; Get address to VA Register
U 1377, 0000,003C,0180,C800,0000,1378  ;5026  D[LONG]_CACHE.P  ; Get the vector
U 1378, 0001,003C,0180,0980,0000,10C8  ;5027  RC[0]_D   ; ... Save for decode
U 10C8, 0000,003D,0180,0800,0000,1283  ;5028 =00 CALL[DECODE]  ; Go decode the retreived vector
U 10C9, 0000,003C,0180,0800,0000,119A  ;5029  J/TST30.MAX.ERR3 ; Maximum amount of errors occured call error
U 10CA, 0000,003C,0180,0800,0000,10CB  ;5030  nop   ; An acceptable amount of errors occured cont..
U 10CB, 0000,003C,0180,0800,0000,137A  ;5031  J/TST30.BANK.CONT3 ; No errors continue test
          ;5032 =0
          ;5033 TST30.MAX.ERR3:
U 119A, 0000,003D,0180,0800,0000,1237  ;5034  CALL[ERDAT]  ; Get logout data
U 119B, 0000,003C,0180,0800,0000,1379  ;5035  nop
```

```
            ;5036  .PAGE
            ;5037 ;
            ;5038 ;////////////////////////////////////////////////////////////////////////////////
            ;5039 ;
U 1379, 0000,003D,6180,0800,0084,704C  ;5040  ERROR1[.F]  ; Log the error
            ;5041 ;
            ;5042 ;////////////////////////////////////////////////////////////////////////////////
            ;5043 ;
            ;5044 ;  I/O BASE ADDRESS OF CURRENTLY TESTED MS780E
            ;5045 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;5046 ;  Received Memory Data
            ;5047 ;  Expected Memory Data
            ;5048 ;  Received Memory Data
            ;5049 ;  Expected Memory Data
            ;5050 ;  Received Memory Data
            ;5051 ;  Expected Memory Data
            ;5052 ;  Received Memory Data
            ;5053 ;  Expected Memory Data
            ;5054 ;  Received Memory Data
            ;5055 ;  Expected Memory Data
            ;5056 ;  Received Memory Data
            ;5057 ;  Expected Memory Data
            ;5058 ;  FFFFFFFF
            ;5059 ;
            ;5060 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5061 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;5062 ;        be output on the error then the last longword wrote will be the
            ;5063 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5064 ;
            ;5065 ;////////////////////////////////////////////////////////////////////////////////
            ;5066 ;
            ;5067 TST30.BANK.CONT3:
U 137A, 0810,0038,0180,0920,0000,137B  ;5068  D_RC[04]          ; Move pattern count to the D register
U 137B, 0019,0014,0580,09A0,0000,137C  ;5069  RC[04]_D+K[.1]  ; Point to the next pattern
U 137C, 0810,0038,0180,0958,0000,137D  ;5070  D_RC[0B]          ; Move Bank count to the D register
U 137D, 0019,0014,0580,09D8,0000,137E  ;5071  RC[0B]_D+K[.1]  ; Point to the next Bank
            ;5072 ;
            ;5073 ; See if we are done with all the Banks on this board
            ;5074 ;
U 137E, 0818,0038,1180,0800,0000,137F  ;5075  D_K[.4]   ; Move Max bank number to D register
            ;5076  ALU_D.XOR.RC[0B], ; Compare the bank counter to max bank
U 137F, 0011,0020,0180,0958,0010,1380  ;5077   CLK.UBCC ; ... if they are equal we are finished
U 1380, 0000,013C,0180,0800,0000,119C  ;5078  Z?    ; Are they equal ?
U 119C, 0000,003C,0180,0800,0000,1198  ;5079 =0 J/TST30.BANK.LOOP5 ; No continue READING encoded pattern of 0's
            ;5080      ; Yes continue with next part of te
            ;5081 ;
            ;5082 ; Set up to point to next board
            ;5083 ;
U 119D, 0010,0038,01C0,0938,0000,1381  ;5084  Q_RC[07]  ; Move Displacement to Q
U 1381, 0810,0038,0180,0940,0000,1382  ;5085  D_RC[08]  ; Move board pointer to D
U 1382, 001D,0014,0180,09C0,0000,1383  ;5086  ALU_D+Q,RC[08]_ALU ; Point to next board
U 1383, 0810,0038,0180,0940,0000,1384  ;5087  D_RC[08]  ; Move board pointer to D register
            ;5088  ALU_D.XOR.RC[06], ; Compare the board counter to max address
U 1384, 0011,0020,0180,0930,0010,1385  ;5089   CLK.UBCC ; ... if they are equal we are finished
U 1385, 0000,013C,0180,0800,0000,119E  ;5090  Z?    ; Are they equal ?
```

```
U 119E, 0000,003C,0180,0800,0000,1376  ;5091 =0 J/TST30.BANK.LOOP6 ; No continue reading encoded pattern of 0's
          ;5092     ; ... Yes continue with next part of test
          ;5093  ;
          ;5094  ; This is the last section of this test and here we check the value
          ;5095  ; of state.  IF state is not equal to zero then get data for logout
          ;5096  ; and call error1, Else return to the caller.
          ;5097  ;
U 119F, 0000,003C,0180,0800,1480,1386  ;5098  SC_STATE  ; Move STATE to the SC
U 1386, 0018,0038,1DC0,0800,0000,1387  ;5099  Q_SC    ; Put SC (STATE) in Q
U 1387, 0001,203C,0180,0800,0010,1388  ;5100  ALU_Q,CLK.UBCC  ; Move Q (State) into the ALU for check
U 1388, 0000,013C,0180,0800,0000,11A0  ;5101  Z?    ; Is state 0 ?
U 11A0, 0000,003C,0180,0800,0000,11A2  ;5102 =0 J/T.30.STATE.NE.ZERO1 ; No, get the data for logout and call an error
U 11A1, 0000,003E,0180,0800,0000,0001  ;5103  RETURN1   ; Yes, return to the caller
          ;5104  ;
          ;5105  ; Come here after checking the value of state and finding that it is
          ;5106  ; non zero.  Get the data for the logout and call an error.
          ;5107  ;
          ;5108 =0
          ;5109 T.30.STATE.NE.ZERO1:
U 11A2, 0000,003D,0180,0800,0000,1237  ;5110  CALL[ERDAT]  ; Get the data for logout
U 11A3, 0000,003C,0180,0800,0000,1389  ;5111  NOP
```

```
            ;5112  .PAGE
            ;5113 ;
            ;5114 ;////////////////////////////////////////////////////////////////////////////////
            ;5115 ;
            ;5116 ; This will call an error from state no equal to zero.
            ;5117 ;
U 1389, 0000,003D,6180,0800,0084,704C  ;5118  ERROR1[.F]
U 138A, 0000,003E,0180,0800,0000,0001  ;5119  RETURN1
            ;5120 ;
            ;5121 ;////////////////////////////////////////////////////////////////////////////////
            ;5122 ;
            ;5123 ; ERROR LOGOUT:
            ;5124 ;
            ;5125 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5126 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;5127 ;  Received Memory Data
            ;5128 ;  Expected Memory Data
            ;5129 ;  Received Memory Data
            ;5130 ;  Expected Memory Data
            ;5131 ;  Received Memory Data
            ;5132 ;  Expected Memory Data
            ;5133 ;  Received Memory Data
            ;5134 ;  Expected Memory Data
            ;5135 ;  Received Memory Data
            ;5136 ;  Expected Memory Data
            ;5137 ;  Received Memory Data
            ;5138 ;  Expected Memory Data
            ;5139 ;  FFFFFFFF
            ;5140 ;
            ;5141 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;5142 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;5143 ;        be output on the error then the last longword wrote will be the
            ;5144 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;5145 ;
            ;5146 ;////////////////////////////////////////////////////////////////////////////////
            ;5147 ;
            ;5148 TEST.30.EXIT:
U 138B, 0000,003C,0180,0800,0000,11A4  ;5149  NOP    ; EXIT TEST
            ;5150
            ;5151
```

```
            ;5152  .PAGE "TEST 1BA RESERVED"
            ;5153 =0
U 11A4, 0000,003D,0180,0800,0000,1087  ;5154 T1BA: NEWTST
U 11A5, 0000,003C,0180,0800,0000,138C  ;5155  NOP
            ;5156
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 1155
ESKAR49.MCR                    MICRO2  1P(00)    26-JUL-85  17:24:12          Page  13

            ;5157  .PAGE "DUMMY NEWTST"
U 138C, 0000,003D,0180,0800,0000,1087  ;5158 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 1156
ESKAR49.MCR                    MICRO2  1P(00)    26-JUL-85  17:24:12          Page  14
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1931: 1910 1919= 1920= 2766= 2767= 2768= 2771=
U 1008  2422= 2424= 2426= 2431= 2272= 2273= 2274= 2275=
U 1010  2445= 2447= 2448= 2452= 2468= 2469= 2470= 2478=
U 1018  2480= 2482= 2484= 2485= 2411= 2412= 2413= 2414=
U 1020  2529= 2530= 2531= 2533= 2577= 2578= 2579= 2581=
U 1028  2863= 2864= 2865= 2866= 2867= 2868= 2869= 2870=
U 1030  3103= 3104= 3111= 3113= 4094= 4095= 4096= 4098=
U 1038  1921= 1922= 1911 3142= 1974= 1976= 1912 3143=
U 1040  4100= 4101= 4104= 4107= 1987= 1988= 1913 3849=
U 1048  4112= 4114= 4116= 4119= 2048= 2049= 2030: 3853=
U 1050  4286= 4288= 4289= 1915 2094= 2095= 1916 3944=
U 1058  2219= 2220= 2076: 1923 2400= 2401= 2110: 3947=
U 1060  4294= 4295= 4296= 1924 4298= 4300= 4303= 4305=
U 1068  4343= 4345= 4347= 1925 2403= 2408= 3228= 3229=
U 1070  4396= 4397= 4398= 1926 4409= 4410= 4411= 1927
U 1078  4437= 4439= 4440= 1928 3298= 3299= 3300= 3301=
U 1080  4453= 4455= 4456= 1929 4486= 4487= 4488= 1963
U 1088  4502= 4503= 4504= 1964 3388= 3390= 3391= 3394=
U 1090  3535= 3536= 3537= 3538= 3539= 3540= 3541= 3542=
U 1098  3543= 3544= 3545= 3546= 3547= 3548= 3549= 3552=
U 10A0  4530= 4531= 4536= 1965 4608= 4609= 4613= 1966
U 10A8  4634= 4635= 4636= 4637= 2612= 2617= 3835= 3836=
U 10B0  4702= 4703= 4704= 4705= 4826= 4827= 4832= 1967
U 10B8  4916= 4917= 4921= 1968 2622= 2627= 3931= 3932=
U 10C0  4944= 4945= 4946= 4947= 2632= 2633= 1969 3996=
U 10C8  5028= 5029= 5030= 5031= 2678= 2679= 1970 3997=
U 10D0  2842= 2843= 2851= 2852= 2853= 2855= 2858= 2859=
U 10D8  2900= 2901= 2902= 2903= 2910= 2911= 2917= 2919=
U 10E0  2922= 2923= 2942= 2943= 2945= 2946= 3037= 3038=
U 10E8  3041= 3042= 3053= 3054= 3057= 3058= 3381= 3383=
U 10F0  3409= 3412= 3415= 3417= 3486= 3488= 3491= 3493=
U 10F8  3494= 3496= 3608= 3609= 3613= 3614= 3628= 3630=
U 1100  1896: 1897: 1898: 1899: 1900: 1901: 1902: 1903:
U 1108  1904: 1971 3688= 3689= 1905: 1906: 1907: 1908:
U 1110  3693= 3694= 3696= 3697= 3708= 3709= 3721= 3722=
U 1118  3785= 3787= 3794= 3795= 3802= 3803= 3810= 3811=
U 1120  3816= 3819= 3843= 3845= 3858= 3862= 3865= 3866=
U 1128  3872= 3875= 3897= 3899= 3906= 3907= 3912= 3915=
U 1130  3939= 3941= 3951= 3952= 3954= 3957= 3970= 3972=
U 1138  3981= 3984= 3989= 3990= 4013= 4015= 4036= 4038=
U 1140  4283= 4284= 4351= 4352= 4365= 4367= 4388= 4389=
U 1148  4400= 4402= 4413= 4415= 4429= 4430= 4445= 4446=
U 1150  4461= 4475= 4477= 4478= 4493= 4494= 4509= 4510=
U 1158  4517= 4518= 4519= 4520= 4525= 4526= 4566= 4568=
U 1160  4572= 4573= 4579= 4580= 4585= 4590= 4640= 4641=
U 1168  4646= 4674= 4676= 4677= 4692= 4694= 4698= 4699=
U 1170  4708= 4709= 4714= 4742= 4754= 4761= 4765= 4766=
U 1178  4773= 4774= 4781= 4782= 4813= 4814= 4815= 4816=
U 1180  4822= 4823= 4862= 4866= 4873= 4875= 4883= 4884=
U 1188  4887= 4888= 4893= 4898= 4950= 4951= 4956= 4984=
U 1190  4986= 4987= 5001= 5006= 5013= 5015= 5016= 5017=
U 1198  5024= 5025= 5034= 5035= 5079= 5084= 5091= 5098=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   5102= 5103= 5110= 5111= 5154= 5155= 1972 1973
U 11A8   1977 1978 1979 1980 1981 1982 1983 1984
U 11B0   1985 1986 2005 2028 2029 2074 2075 2158
U 11B8   2159 2160 2179 2181 2208 2209 2210 2211
U 11C0   2212 2213 2214 2215 2217 2218 2265 2266
U 11C8   2267 2269 2271 2409 2410 2435 2455 2456
U 11D0   2457 2464 2607 2656 2680 2681 2682 2683
U 11D8   2702 2703 2704 2705 2724 2725 2726 2727
U 11E0   2755 2757 2758 2759 2761 2762 2763 2764
U 11E8   2765 2772 2773 2774 2775 2802 2803 2804
U 11F0   2844 2845 2846 2847 2856 2857 2861 2862
U 11F8   2875 2876 2878 2879 2880 2882 2887 2888
U 1200   2889 2891 2896 2898 2899 2907 2908 2912
U 1208   2913 2914 2915 2916 2924 2925 2926 2928
U 1210   2929 2930 2931 2932 2933 2938 2940 2941
U 1218   2944 2969 2970 2971 2972 3001 3002 3004
U 1220   3005 3006 3033 3035 3036 3040 3044 3045
U 1228   3046 3047 3049 3050 3051 3052 3055 3056
U 1230   3097 3098 3099 3100 3101 3115 3117 3134
U 1238   3135 3136 3137 3138 3139 3140 3141 3145
U 1240   3169 3170 3171 3172 3173 3174 3222 3223
U 1248   3224 3225 3226 3227 3294 3295 3296 3297
U 1250   3306 3309 3310 3313 3384 3385 3386 3399
U 1258   3400 3402 3405 3407 3408 3413 3422 3478
U 1260   3480 3481 3482 3484 3485 3530 3533 3601
U 1268   3602 3604 3606 3607 3616 3618 3619 3620
U 1270   3621 3623 3625 3680 3681 3682 3684 3686
U 1278   3687 3699 3700 3704 3706 3710 3712 3713
U 1280   3714 3716 3718 3763 3764 3765 3766 3767
U 1288   3769 3770 3771 3772 3773 3774 3775 3777
U 1290   3778 3780 3781 3783 3784 3788 3789 3790
U 1298   3792 3793 3796 3797 3800 3801 3804 3805
U 12A0   3806 3808 3809 3813 3814 3821 3822 3824
U 12A8   3827 3828 2136: 3829 3831 3832 3833 3834
U 12B0   3839 3840 3848 3855 3857 3863 3867 3868
U 12B8   3869 3870 3871 3876 3878 3879 3881 3882
U 12C0   3883 3884 3885 3886 3887 3889 3890 3892
U 12C8   3893 3895 3896 3900 3901 3902 3904 3905
U 12D0   3909 3910 3917 3918 3920 3923 3924 3925
U 12D8   3927 3928 3929 3930 3935 3936 3943 3949
U 12E0   3950 3953 3958 3959 3961 3962 3964 3966
U 12E8   3968 3974 3976 3979 3987 3988 3993 3995
U 12F0   4008 4010 4011 4018 4021 4024 4025 4027
U 12F8   4028 4029 4030 4031 4033 4035 4162 4315
U 1300   4328 4330 4331 4333 4335 4337 4338 4339
U 1308   4340 4353 4354 4355 4356 4378 4379 4380
U 1310   4382 4384 4386 4387 4427 4428 4441 4442
U 1318   4443 4457 4458 4459 4476 4482 4490 4491
U 1320   4506 4507 4521 4522 4527 4528 4529 4559
U 1328   4560 4561 4562 4564 4565 4569 4574 4575
U 1330   4576 4577 4578 4581 4583 4584 4675 4678
U 1338   4679 4680 4681 4685 4686 4687 4688 4690
U 1340   4691 4695 4700 4701 4743 4747 4748 4749
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1158
ESKAR49.MCR                    MICRO2  1P(00)    26-JUL-85  17:24:12           Page  14

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4750 4752 4753 4762 4763 4764 4817 4819
U 1350  4824 4825 4853 4854 4858 4860 4861 4867
U 1358  4868 4869 4871 4872 4876 4879 4885 4886
U 1360  4889 4891 4892 4941 4942 4943 4985 4988
U 1368  4989 4990 4991 4992 4993 4997 4999 5000
U 1370  5007 5008 5009 5011 5012 5018 5021 5026
U 1378  5027 5040 5068 5069 5070 5071 5075 5077
U 1380  5078 5085 5086 5087 5089 5090 5099 5100
U 1388  5101 5118 5119 5149 5158
U 1390 - 13EF Unused
U 13F0  2120: 2121: 2122: 2123: 2124: 2125: 2126: 2127:
U 13F8  2128: 2129: 2130: 2131: 2132: 2133: 2134: 2135:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR49	925
Used	925
Remaining	

Total microwords used in memory U: 925
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1160
ESKAR49.MCR                    MICRO2  1P(00)    26-JUL-85  17:24:12     Page  14


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR49.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR4A.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 1161

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1852 Micro Trap Handler and LSI-11 Support Routines
; 1853   Micro Trap Handler
; 1933 Micro Monitor Interface Routines
; 1934   Newtest Routine
; 1990   Error Loop Routine
; 2007   Error 1 Routine
; 2032   ERROR1 WITH PARAMETER
; 2053   Error 2 Routine
; 2079   ERROR 2 WITH PARAMETER
; 2098   Micro-Monitor Call
; 2112   Reserved Control Store
; 2139   Set Argument Count
; 2163   Increment Subtest Number
; 2184 Diagnostic Specific Subroutines
; 2185   Select Controller
; 2223   64K OR 256K CHIPS
; 2278   START TEST
; 2488   SELECT LOWER CONTROLLER
; 2536   SELECT UPPER CONTROLLER
; 2585   SBI Unjam Routine
; 2636   RESET SUBTEST NUMBER
; 2658   Initialize the SBI
; 2686   Enable Cache
; 2708   Disable Cache Routine
; 2730   Check for Interrupt Routine
; 2778   Set Trap Mask
; 2807   FIND MS780-E MEMORY
; 2949   Generate IO Address
; 2975   Set Starting Address
; 3009   ENABLE NEXT MS780-E
; 3061   GET MEMORY SIZE
; 3120   ERDAT ROUTINE
; 3149   Set Diagnostic Mode Routine
; 3178   Check for Odd or Even TR
; 3234   RAS ADDRES GENREATOR
; 3293   CAS ADDRESS GENERATOR
; 3357   REED MULLER ENCODING ROUTINE
; 3433   REED-MULLER VECTOR GENERATION ROUTINE
; 3486   FIRST ORDER REED-MULLER ENCODE
; 3567   SECOND ORDER REED-MULLER ENCODING ROUTINE
; 3658   REED-MULLER DECODE ROUTINE
; 3974   FIND CONTROLLERS
; 4055   DELAY
; 4094 TEST 1BB COLUMN/ROW SELECT ADDRESSING
; 4670 TEST 1BC RESERVED
; 4675 DUMMY NEWTST

```
;I  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR4A
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ; MODIFIED BY:
;1836  ;
;1837  ;  1.1 March 10, 1983  Paul Hakala
;1838  ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839  ;   the correct memory address when 256k ram chips are
;1840  ;   present.
;1841  ;
;1842  ;   March 10, 1983  Paul Hakala
;1843  ;   Removed a call to to LOAD.TEMP.REG that wasnt
;1844  ;   needed in test 31 RAS/CAS test
;1845  ;
;1846  ;
;1847  ;
;1848  ;*****************************************************************************
;1849  ;
```

```
;1850  .PAGE
;1851
;1852  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1853  .TOC "   Micro Trap Handler"
;1854 ;++
;1855 ; FUNCTIONAL DESCRIPTION:
;1856 ;
;1857 ; This routine is responsible for 'catching' micro-traps
;1858 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1859 ; contains the Trap-Expected Mask. If the specified bit is set in
;1860 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1861 ; an error call is made.
;1862 ;
;1863 ; INPUT PARAMETERS:
;1864 ;
;1865 ; R[0F] - Expected Trap Mask
;1866 ;  Trap Code Function
;1867 ;  ---------      --------
;1868 ;       0  System Init
;1869 ;       1  Data Unaligned
;1870 ;       2  Cross Page
;1871 ;       3  TB Modify
;1872 ;       4  TB Protection
;1873 ;       6  TB Miss
;1874 ;       7  TB Parity
;1875 ;       8  Cache Parity
;1876 ;       9  Reserved Floating Operand
;1877 ;       C  Read Data Substitute
;1878 ;       D  Time out
;1879 ;       F  Control Store Parity Error
;1880 ;      10  Odd Address
;1881 ;
;1882 ; OUTPUT PARAMETERS:
;1883 ;
;1884 ; R[0F] - Mask bit is cleared.
;1885 ;
;1886 ; DATA: Micro PC of Micro Trap
;1887 ;  Trap Code (See Above)
;1888 ;  Fault Status Register
;1889 ;  SBI Error Register
;1890 ;  Timeout Address Register
;1891 ;  Maintenance Register
;1892 ;  Cache Parity Register
;1893 ;  TB Error Register 1
;1894 ;  TB Error Register 0
;1895 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1896 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1897 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1898 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1899 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1900 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1901 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1902 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1903 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1904 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000,003C,8580,0800,0084,7001  ;1905 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000,003C,8980,0800,0084,7001  ;1906 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000,003C,6580,0800,0084,7001  ;1907 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000,003C,6180,0800,0084,7003  ;1908 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1909
U 1001, 0000,003C,81F0,2C00,0000,103A  ;1910 trph0: q_id[ustack]  ; Get upc of trap
U 103A, 0C00,003C,01E0,0800,0000,103E  ;1911  q_d,d_q   ; Setup to put it back on stack
U 103E, 0C00,003C,81E0,3C00,0000,1046  ;1912  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1046, 0000,003C,01C0,0A78,0000,1053  ;1913  q_r[0f]   ; Get the Expected Trap Mask
          ;1914  alu_q.andnot.mask,
U 1053, 0001,2024,0180,0800,0010,1056  ;1915   clk.ubcc ; Was trap expected?
U 1056, 0000,013C,0180,0800,0000,1002  ;1916  z?
          ;1917 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1918      alu_q.and.mask, ; ...
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1919      return0  ; ...
U 1003, 0018,0038,1D80,09E8,0000,1038  ;1920 trph1: rc[0d]_sc  ; Output the Trap Code
U 1038, 0000,003D,D980,0800,0084,7169  ;1921 =0 setrcf[.9]  ; Set output count
U 1039, 0000,003C,49F0,2C00,0000,105B  ;1922  q_id[tber0]  ; Output all the error registers
U 105B, 0001,203C,4DF0,2DB0,0000,1063  ;1923  rc[6]_q,q_id[tber1] ; ...
U 1063, 0001,203C,65F0,2DB8,0000,10A6  ;1924  rc[7]_q,q_id[sbi.err] ; ...
U 10A6, 0001,203C,6DF0,2DD8,0000,10AE  ;1925  rc[0b]_q,q_id[fault] ; ...
U 10AE, 0001,203C,75F0,2DE0,0000,1109  ;1926  rc[0c]_q,q_id[maint] ; ...
U 1109, 0001,203C,79F0,2DC8,0000,114D  ;1927  rc[9]_q,q_id[parity] ; ...
U 114D, 0001,203C,69F0,2DC0,0000,114E  ;1928  rc[8]_q,q_id[time.addr] ; ...
U 114E, 0001,203C,81F0,2DD0,0000,1000  ;1929  rc[0a]_q,q_id[ustack] ; ...
          ;1930 1000: ERROR1[.C],  ;
U 1000, 0001,203D,8580,09F0,0084,704C  ;1931   rc[0e]_q ; Report the error
```

```
                ;1932  .PAGE
                ;1933  .TOC " Micro Monitor Interface Routines"
                ;1934  .TOC "   Newtest Routine"
                ;1935 ;++
                ;1936 ; FUNCTIONAL DESCRIPTION:
                ;1937 ;
                ;1938 ; This routine Initializes the following registers:
                ;1939 ;  PSL    to 1F
                ;1940 ;  CES    to 0
                ;1941 ;  ACC.CS    to disable accellerator
                ;1942 ;  SIR    to 0
                ;1943 ;  CLK.CS    to stop interval timer
                ;1944 ;  TBER0    to disable the TB
                ;1945 ;  SBI.MAINT to     0
                ;1946 ;  SBI.ERR    to 0
                ;1947 ;  FAULT    to 0
                ;1948 ;  COMP    to 0
                ;1949 ;  RC[0E]    to  0
                ;1950 ;  R[0F]    to 0
                ;1951 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1952 ; A SCOPE call is then made to the Monitor.
                ;1953 ;
                ;1954 ; CALLING CONSTRAINT:
                ;1955 ;
                ;1956 ; =0
                ;1957 ;
                ;1958 ; SIDE EFFECTS:
                ;1959 ;
                ;1960 ; D Reg is destroyed
                ;1961 ; SC is destroyed
                ;1962 ;--
U 114F, 0000,003C,65F8,0800,0084,7150  ;1963 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1150, 0818,0038,8D80,0800,0000,1151  ;1964  d_k[.1f]  ; D=1F
U 1151, 0D00,003C,0180,0800,0000,1152  ;1965  d_dal.sc  ; D=001F0000
U 1152, 0000,003C,3D80,3C00,0000,1153  ;1966  id[psl]_d  ; psl = 001F0000
U 1153, 0818,0038,0580,0800,0000,1154  ;1967  d_k[.1]  ; Clear the CES register
U 1154, 0D00,003C,0180,0800,0000,1155  ;1968  d_dal.sc  ; D = 00010000
U 1155, 0F00,003C,3180,3C00,0000,1156  ;1969  id[ces]_d,d_0  ; ces = 00010000
U 1156, 0000,003C,5D80,3C00,0000,1157  ;1970  id[acc.cs]_d  ; acc.cs = 0
U 1157, 0000,003C,3980,3C00,0000,1158  ;1971  id[sir]_d  ; sir = 0
U 1158, 0000,003C,2980,3C00,0000,1159  ;1972  id[clk.cs]_d  ; clk.cs = 0
U 1159, 0000,003C,4980,3C00,0000,103C  ;1973  id[tber0]_d  ; tber0 = 0
U 103C, 0000,003D,0180,0800,0000,1184  ;1974 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1975  intrpt.strobe,  ; Strobe interrupts
U 103D, 0818,0038,C180,0800,4000,115A  ;1976      d_k[.ffff]  ; Setup to clear SBI error register and
U 115A, 0801,0028,6580,3C00,0000,115B  ;1977  id[sbi.err]_d,d_not.d ; and fault register
U 115B, 0F00,003C,6D80,3C00,0000,115C  ;1978  id[fault]_d,d_0  ; ...
U 115C, 0000,003C,6D80,3C00,0000,115D  ;1979  id[fault]_d  ; fault = 0
U 115D, 0000,003C,7580,3C00,0000,115E  ;1980  id[maint]_d  ; MAINT = 0
U 115E, 0000,003C,6580,3C00,0000,115F  ;1981  id[sbi.err]_d  ; sbi error reg = 0
U 115F, 0000,003C,7180,3C00,0000,1160  ;1982  id[comp]_d  ; comp reg = 0
U 1160, 0000,003C,E580,3C00,0000,1161  ;1983  ID[39]_d  ; Clear code for subroutine START.TEST
U 1161, 0001,003C,0180,09F0,0000,1162  ;1984  rc[0e]_d  ;
U 1162, 0001,003C,0180,0AF8,0000,1163  ;1985  r[0f]_d  ; Clear expected trap mask
U 1163, 0001,003C,0180,09F8,0000,1044  ;1986  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1044, 0000,003D,0180,0800,0000,118E  ;1987 =0 call[disable.cache] ; Disable the cache
U 1045, 0F03,003C,0180,09E8,0000,105E  ;1988  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1989  .PAGE
            ;1990  .TOC "    Error Loop Routine"
            ;1991 ;++
            ;1992 ; FUNCTIONAL DESCRIPTION:
            ;1993 ;
            ;1994 ; This routine is called with the "ERLOOP" macro. The
            ;1995 ; routine calls the Micro Monitor to setup an error loop.
            ;1996 ;
            ;1997 ; CALLING CONSTRAINT:
            ;1998 ;
            ;1999 ; =0
            ;2000 ;
            ;2001 ; SIDE EFFECTS:
            ;2002 ;
            ;2003 ; D Reg - Destroyed
            ;2004 ;--
U 1164, 0818,0038,0980,0800,0000,105E  ;2005 ERLOOP: d_k[.2],j/callcon
```

```
            ;2006  .PAGE
            ;2007  .TOC "   Error 1 Routine"
            ;2008 ;++
            ;2009 ; FUNCTIONAL DESCRIPTION:
            ;2010 ;
            ;2011 ; This routine is used to report an error to the user. This
            ;2012 ; routine is used when you do not wish to continue in the
            ;2013 ; same test after the call to the Monitor.
            ;2014 ;
            ;2015 ; CALLING CONSTRAINT:
            ;2016 ;
            ;2017 ; None
            ;2018 ;
            ;2019 ; INPUTS:
            ;2020 ;
            ;2021 ; rc[0f]<15:8> - Contain the subtest number
            ;2022 ;
            ;2023 ; OUTPUTS:
            ;2024 ;
            ;2025 ; D<15:8> - Subtest number
            ;2026 ; D<7:0> - Error 1 Monitor Code
            ;2027 ;--
U 1165, 0810,0038,0180,0978,0000,1166  ;2028 ERROR1: d_rc[0f]  ; Get the subtest number
U 1166, 0819,0034,4D80,0800,0000,104E  ;2029  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2030 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2031  .PAGE
            ;2032  .TOC "   ERROR1 WITH PARAMETER"
            ;2033 ;++
            ;2034 ; FUNCTIONAL DESCRIPTION:
            ;2035 ;
            ;2036 ; This subroutine takes  as parameter the number of arguments
            ;2037 ; to be printed to the console in the case of an error logout.
            ;2038 ;
            ;2039 ; CALLING CONSTRAINT:
            ;2040 ;
            ;2041 ; =0
            ;2042 ;INPUTS:
            ;2043 ;
            ;2044 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2045 ;--
            ;2046 =0
            ;2047 ERR1.ARG:
U 104C, 0000,003D,0180,0800,0000,1169  ;2048  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 104D, 0000,003C,0180,0800,0000,1165  ;2049  J/ERROR1  ; Go to ERROR1
            ;2050 ;
            ;2051 ;
```

```
            ;2052  .PAGE
            ;2053  .TOC "   Error 2 Routine"
            ;2054 ;++
            ;2055 ; FUNCTIONAL DESCRIPTION:
            ;2056 ;
            ;2057 ; This routine is used  to report an error to the user. This
            ;2058 ; routine is used when you wish to continue in the same test
            ;2059 ;       after the call to the Monitor.
            ;2060 ;
            ;2061 ; CALLING CONSTRAINT:
            ;2062 ;
            ;2063 ; =0
            ;2064 ;
            ;2065 ; INPUTS:
            ;2066 ;
            ;2067 ; rc[0f]<15:8> - Contain the subtest number
            ;2068 ;
            ;2069 ; OUTPUTS:
            ;2070 ;
            ;2071 ; D<15:8> - Subtest number
            ;2072 ; D<7:0> - Error 2 Monitor Code
            ;2073 ;--
U 1167,  0810,0038,0180,0978,0000,1168  ;2074 ERROR2: d_rc[0f]  ; Get the subtest number
U 1168,  0819,0034,4D80,0800,0000,105A  ;2075  d_d.and.k[.ff00] ; ...
U 105A,  0819,0030,D580,0800,0000,105E  ;2076 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2077 ;
```

```
            ;2078  .PAGE
            ;2079  .TOC "   ERROR 2 WITH PARAMETER"
            ;2080 ;**
            ;2081 ; FUNCTIONAL DESCRIPTION:
            ;2082 ;
            ;2083 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2084 ; except that in this case after setting the number of arguments
            ;2085 ; too the console, control is transferred to routine ERROR2.
            ;2086 ;
            ;2087 ; INPUTS:
            ;2088 ;
            ;2089 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2090 ;
            ;2091 ;--
            ;2092 =0
            ;2093 ERR2.ARG:
U 1054, 0000.003D.0180.0800.0000.1169  ;2094  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1055, 0000.003C.0180.0800.0000.1167  ;2095  J/ERROR2  ; Go to ERROR2
            ;2096 ;
```

```
            ;2097  .PAGE
            ;2098  .TOC "   Micro-Monitor Call"
            ;2099 ;*
            ;2100 ; FUNCTIONAL DESCRIPTION:
            ;2101 ;
            ;2102 ; This micro word calls the micro monitor.
            ;2103 ;
            ;2104 ; EXPLICIT INPUTS:
            ;2105 ;
            ;2106 ; D reg must contain valid monitor code.
            ;2107 ;--
            ;2108 105E:
            ;2109 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2110  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2111   .PAGE
            ;2112   .TOC "   Reserved Control Store"
            ;2113 ;**
            ;2114 ; FUNCTIONAL DESCRIPTION:
            ;2115 ;
            ;2116 ; The following 16 locations must be reserved so the monitor
            ;2117 ; can use them to perform various functions that require
            ;2118 ; the execution of micro-code.
            ;2119 ;--
U 13F0, 0000,003C,0180,0800,0000,13F1   ;2120 13F0: nop
U 13F1, 0000,003C,0180,0800,0000,13F2   ;2121 13F1: nop
U 13F2, 0000,003C,0180,0800,0000,13F3   ;2122 13F2: nop
U 13F3, 0000,003C,0180,0800,0000,13F4   ;2123 13F3: nop
U 13F4, 0000,003C,0180,0800,0000,13F5   ;2124 13F4: nop
U 13F5, 0000,003C,0180,0800,0000,13F6   ;2125 13F5: nop
U 13F6, 0000,003C,0180,0800,0000,13F7   ;2126 13F6: nop
U 13F7, 0000,003C,0180,0800,0000,13F8   ;2127 13F7: nop
U 13F8, 0000,003C,0180,0800,0000,13F9   ;2128 13F8: nop
U 13F9, 0000,003C,0180,0800,0000,13FA   ;2129 13F9: nop
U 13FA, 0000,003C,0180,0800,0000,13FB   ;2130 13FA: nop
U 13FB, 0000,003C,0180,0800,0000,13FC   ;2131 13FB: nop
U 13FC, 0000,003C,0180,0800,0000,13FD   ;2132 13FC: nop
U 13FD, 0000,003C,0180,0800,0000,13FE   ;2133 13FD: nop
U 13FE, 0000,003C,0180,0800,0000,13FF   ;2134 13FE: nop
U 13FF, 0000,003C,0180,0800,0000,12AA   ;2135 13FF: nop
U 12AA, 0000,003C,0180,0800,0000,1169   ;2136 12AA: nop ; This location is permanently blasted in the FPLA
            ;2137   ; to cause a micro ECO to location 12AA.
```

```
            ;2138  .PAGE
            ;2139  .TOC "   Set Argument Count"
            ;2140 ;**
            ;2141 ; FUNCTIONAL DESCRIPTION:
            ;2142 ;
            ;2143 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2144 ; to the console.
            ;2145 ;
            ;2146 ; CALLING CONSTRAINT:
            ;2147 ;
            ;2148 ; =0
            ;2149 ;
            ;2150 ; INPUTS:
            ;2151 ;
            ;2152 ; SC - Contains new value of argument count
            ;2153 ;
            ;2154 ; SIDE EFFECTS:
            ;2155 ;
            ;2156 ; Q is destroyed
            ;2157 ;--
U 1169, 0010,0038,01C0,0978,0000,116A  ;2158 SETRCF: q_rc[0f]  ; Get the argument count
U 116A, 0019,2034,4DC0,0800,0000,116B  ;2159  q_q.and.k[.ff00] ; ...
U 116B, 0019,2032,1D80,09F8,0000,0001  ;2160  rc[0f]_q.or.sc,return1 ; Set new argument count and return
            ;2161
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1177
ESKAR4A.MCR                    MICRO2  1P(00)    26-JUL-85  17:26:47                   Page   6

            ;2162  .PAGE
            ;2163  .TOC "   Increment Subtest Number"
            ;2164 ;++
            ;2165 ; FUNCTIONAL DESCRIPTION:
            ;2166 ;
            ;2167 ; This routine is used to increment the subtest number
            ;2168 ; in RC[0F]<15:8>.
            ;2169 ;
            ;2170 ; CALLING CONSTRAINT:
            ;2171 ;
            ;2172 ; =0
            ;2173 ;
            ;2174 ; SIDE EFFECTS:
            ;2175 ;
            ;2176 ; D register is destroyed
            ;2177 ;--
            ;2178 subtst:
U 116C, 0810,0038,4D80,0978,0000,116D  ;2179  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2180  rc[0f]_d+k[.ff00], ; Increment and return
U 116D, 0019,4016,4D80,09F8,0000,0001  ;2181      word,return1 ; (Subtest number is negative)
            ;2182
```

```
            ;2183  .PAGE
            ;2184  .TOC " Diagnostic Specific Subroutines"
            ;2185  .TOC "   Select Controller"
            ;2186 ;++
            ;2187 ; FUNCTIONAL DESCRIPTION:
            ;2188 ;
            ;2189 ; This routine is used to put the memory system into the
            ;2190 ; specified configuration with respect to controller select.
            ;2191 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2192 ; a RETURN2 is made.
            ;2193 ;
            ;2194 ; CALLING CONSTRAINT:
            ;2195 ;
            ;2196 ; =00
            ;2197 ;
            ;2198 ; INPUTS:
            ;2199 ;
            ;2200 ; d - Contains value to write to bits<2:0> of Register A
            ;2201 ; rc[0e] - Address of Register A
            ;2202 ;
            ;2203 ; SIDE EFFECTS:
            ;2204 ;
            ;2205 ; sc,d,va are destroyed
            ;2206 ;--
            ;2207 SELECT.CNTRLR:
U 116E, 0010.0038.0180.0970.0284.716F  ;2208  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 116F, 0801.001C.0180.0800.0000.1170  ;2209  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1170, 0000.003C.0180.A800.0000.1171  ;2210  cache.p_d[long]  ; Write the configuration Register
U 1171, 0818.0038.5D80.0800.0000.1172  ;2211  d_k[.7]  ; Get Misconfiguration bits
U 1172, 0000.003C.61F8.0800.0084.7173  ;2212  sc_k[.F],q_0  ; ...
U 1173, 0D00.003C.0180.0800.0000.1174  ;2213  d_dal.sc  ; ... D = x38000
U 1174, 0000.003C.01E0.0800.0000.1175  ;2214  q_d  ; Save mask
U 1175, 0000.003C.0180.C800.0000.1176  ;2215  d[long]_cache.p  ; Read CNFG A Reg and
            ;2216  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1176, 001D.2034.0180.0800.0010.1177  ;2217  clk.ubcc ; ...
U 1177, 0000.013C.0180.0800.0000.1058  ;2218  z?  ; Branch if no
U 1058, 0000.003E.0180.0800.0000.0001  ;2219 =0 RETURN1  ; Bad configuration
U 1059, 0000.003E.0180.0800.0000.0002  ;2220  RETURN2  ; Configuration ok
            ;2221
```

```
                ;2222  .PAGE
                ;2223  .TOC "   64K OR 256K CHIPS"
                ;2224 ;*************************************************************
                ;2225 ;++
                ;2226 ;
                ;2227 ; Functional Description:
                ;2228 ; ----------------------
                ;2229 ; This subroutine is used to find the type of chips
                ;2230 ; on the MS780-E/H arrays.  The RETURN modes are as
                ;2231 ; follows:
                ;2232 ;
                ;2233 ; RETURN1 = Error. Mixed 64K and 256K arrays
                ;2234 ;
                ;2235 ; RETURN2 = 64K chips on the array
                ;2236 ;
                ;2237 ; RETURN3 = 256K chips on the array
                ;2238 ;
                ;2239 ; Calling Constraint: =00
                ;2240 ; ------------------
                ;2241 ;
                ;2242 ;
                ;2243 ; Inputs:
                ;2244 ; ------
                ;2245 ;
                ;2246 ; RC[0E] must hold the I/O base address of the MS780-E/H
                ;2247 ; currently under test
                ;2248 ;
                ;2249 ; Outputs:
                ;2250 ; -------
                ;2251 ;
                ;2252 ; NO specific outputs. (See RETURN modes above)
                ;2253 ;
                ;2254 ;
                ;2255 ; Side Effects:
                ;2256 ; ------------
                ;2257 ;
                ;2258 ; The contents of the following registers will be lost:
                ;2259 ;
                ;2260 ; D
                ;2261 ;
                ;2262 ;--
                ;2263 ;*************************************************************
                ;2264 64K.OR.256K:
U 1178, 0010,0038,0180,0970,0200,1179  ;2265  VA_RC[0E]    ; Point to CNFG Register A
U 1179, 0000,003C,0180,C800,0000,117A  ;2266  D[LONG]_CACHE.P    ; Read the register
U 117A, 0200,003C,0180,0800,0000,117B  ;2267  D_D.RIGHT2    ; Shift received contents two bits
                ;2268          ; ...to the right
U 117B, 0600,003C,0180,0800,0000,117C  ;2269  D_D.RIGHT    ; Shift if right one more time for a
                ;2270          ; total shift right of 3 bits
U 117C, 0000,193C,0180,0800,0000,100C  ;2271  D1-0?     ; Branch on the RAM Type
U 100C, 0000,003E,0180,0800,0000,0001  ;2272 =1100 RETURN1     ; Error: both 64K and 256K RAMs present
U 100D, 0000,003E,0180,0800,0000,0002  ;2273  RETURN2     ; 64K RAMs found
U 100E, 0000,003E,0180,0800,0000,0003  ;2274  RETURN3     ; 256K RAMs found
U 100F, 0000,003E,0180,0800,0000,0001  ;2275  RETURN1     ; Error: missing arrays
                ;2276
```

```
;2277  .PAGE
;2278  .TOC "   START TEST"
;2279 ;*************************************************************
;2280 ;**
;2281 ;
;2282 ; Functional Description:
;2283 ; ---------------------
;2284 ;
;2285 ;        This subroutine will be called by all tests that  check  the
;2286 ;    controllers,  or  those  tests  that  have  to switch between the
;2287 ;    controllers when executing.  Its  main  functions  are:  select
;2288 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2289 ;    test that called it, call out an error  if  neither  one  of  the
;2290 ;    controllers can be configured properly, if all the controllers on
;2291 ;    a MS780-E/H were configured and tested, it should select the next
;2292 ;    MS780-E/H on the SBI and repeat the above sequence.
;2293 ;    A test making use of this  subroutine  should  be  configured  as
;2294 ;    follows:
;2295 ;
;2296 ;
;2297 ;       Begin TEST.XX
;2298 ;        :
;2299 ;        :   Call NEWTST
;2300 ;        :   Call FIND.MS780E
;2301 ;        :   IF No MS780-E's on the SBI
;2302 ;        :     THEN
;2303 ;        :         Skip to End of TEST.XX
;2304 ;        :     ELSE
;2305 ;        :
;2306 ;        : RESTART.TEST.XX:
;2307 ;        :
;2308 ;        :   Call START TEST
;2309 ;        :   IF Return1 from START.TEST
;2310 ;        :     THEN
;2311 ;        :         Skip to End of TEST.XX
;2312 ;        :     ELSE.
;2313 ;        :
;2314 ;        :
;2315 ;        :
;2316 ;        :
;2317 ;        :   Body of TEST.XX
;2318 ;        :
;2319 ;        :
;2320 ;        :
;2321 ;        :   Jump RESTART.TEST.XX
;2322 ;        :
;2323 ;       End TEST.XX
;2324 ;
;2325 ;
;2326 ;          This subroutine makes use of a pointer  in  ID  Register  39
;2327 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2328 ;    should be passed when the subroutine is called a second, third or
;2329 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2330 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2331 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2332 ;    Register 39 can be interpreted as follows:
;2333 ;
;2334 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2335 ;               called the first time.  When this code is encountered
;2336 ;               this subroutine will try to select the lower controller.
;2337 ;               If the lower controller is misconfigured, the pointer in
;2338 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2339 ;               is re-entered.
;2340 ;               If, however there is no misconfiguration, the value in
;2341 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2342 ;               calling test.
;2343 ;
;2344 ;    b01  -     This value signifies that an attempt at configuring the
;2345 ;               lower controller failed, and the subroutine will attempt
;2346 ;               to select the upper controller.
;2347 ;               If the upper controller is also misconfigured execution
;2348 ;               stops with a call to ERROR1.
;2349 ;               If there is no misconfiguration on this controller, then
;2350 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2351 ;               made to the calling test.
;2352 ;
;2353 ;    b10  -     This value signifies that the lower controller was selected
;2354 ;               previously and the test was executed once.  If the
;2355 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2356 ;               39 is loaded with b11 and an attempt is made to select the
;2357 ;               upper controller.
;2358 ;               If there is a misconfiguration on this controller, this
;2359 ;               subroutine is just re-entered.
;2360 ;               Otherwise, a Return2 will be made to the calling test.
;2361 ;
;2362 ;    b11   -    This code identifies the cases in which the test has
;2363 ;               been executed at least once (i.e., at least one of the
;2364 ;               controllers on the MS780-E unit under test could be
;2365 ;               configured properly).  When this code is detected the
;2366 ;               subroutine clears ID Reg 39 and makes a call to
;2367 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2368 ;               tested then the subroutine executes a Return1.
;2369 ;               Otherwise the subroutine is re-entered.
;2370 ;
;2371 ;
;2372 ; Calling Constraint: =00
;2373 ; ------------------
;2374 ;
;2375 ;
;2376 ; Inputs:
;2377 ; ------
;2378 ;
;2379 ; ID Register 39 must be cleared by NEWTST
;2380 ;
;2381 ;
;2382 ; Outputs:
;2383 ; -------
;2384 ;
;2385 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2386 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2387 ;         to be tested
            ;2388 ;
            ;2389 ; Side Effects:
            ;2390 ; ------------
            ;2391 ;
            ;2392 ; Contents of the following regiters will be destroyed:
            ;2393 ;
            ;2394 ; D, Q
            ;2395 ;
            ;2396 ;--
            ;2397 ;*************************************************************
            ;2398 =0
            ;2399 START.TEST:
U 105C, 0000,003D,0180,0800,0000,1185  ;2400  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 105D, 0000,003C,0180,0800,0000,106C  ;2401  NOP    ; ...tests that have more than one
            ;2402 =0      ; ...subtest.)
U 106C, 0000,003D,0180,0800,0000,1164  ;2403  ERLOOP    ; Set up the error loop for the
            ;2404       ; ...eventuality that the MS780-E/H
            ;2405       ; ...currently under test has no
            ;2406       ; ...Controllers
            ;2407 RE.ENTRY:    ; Re entry point for this routine
U 106D, 0000,003C,E5F0,2C00,0000,117D  ;2408  Q_ID[39]   ; Get the code
U 117D, 0C00,003C,0180,0800,0000,117E  ;2409  D_Q    ; Put it in the D Register
U 117E, 0000,193C,0180,0800,0000,101C  ;2410  DI-0?    ; Branch on the code
U 101C, 0000,003C,0180,0800,0000,1008  ;2411 =1100 J/STRT.CASE00   ; Code is 00
U 101D, 0000,003C,0180,0800,0000,1010  ;2412  J/STRT.CASE01   ; Code is 01
U 101E, 0000,003C,0180,0800,0000,1183  ;2413  J/STRT.CASE10   ; Code is 10
U 101F, 0000,003C,0180,0800,0000,1017  ;2414  J/STRT.CASE11   ; Code is 11
            ;2415 ;
            ;2416 ;
            ;2417 ; At this point the code in ID[39] was 00
            ;2418 ;
            ;2419 ;
            ;2420 =00
            ;2421 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1020  ;2422  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2423  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,117F  ;2424   D_K[.1]   ; Set up the code for re entry to this
            ;2425       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2426  D_K[.2]    ; Set up the code for a next call to
            ;2427       ; ...START.TEST indicating that the
            ;2428       ; ...Lower Controller was configured
            ;2429       ; ...( 10 )
            ;2430  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2431   RETURN2   ; Let the test know that the Lower
            ;2432       ; ...has been configured
            ;2433 STRT.LOW.MIS:
            ;2434  ID[39]_D,    ; Save code in ID[39] signifing that
U 117F, 0000,003C,E580,3C00,0000,106D  ;2435   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2436       ; ...end re enter this subroutine
            ;2437
            ;2438 ;
            ;2439 ;
            ;2440 ; At this point the code is 01
            ;2441 ;
```

```
            ;2442 ;
            ;2443 =00
            ;2444 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1024  ;2445  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2446  J/STRT.UPR.MIS,   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,1180  ;2447   D_0   ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2448  D_K[.3]    ; Upper controller was configured
            ;2449       ; ...thus the code must be changed
            ;2450       ; ...accordingly ( 11 )
            ;2451  ID[39]_D,   ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2452   RETURN2   ; Let the test know that the Upper
            ;2453       ; ...Controller has been configured
            ;2454 STRT.UPR.MIS:
U 1180, 0000,003C,E580,3C00,0000,1181  ;2455  ID[39]_D   ; Save the Code ( 00 )
U 1181, 0018,0038,0D80,09E8,0000,1182  ;2456  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 1182, 0000,003D,0980,0800,0084,704C  ;2457  ERROR1[.2]   ; Flag the error.
            ;2458 ;
            ;2459 ;
            ;2460 ; At this point the code is 10
            ;2461 ;
            ;2462 ;
            ;2463 STRT.CASE10:
U 1183, 0818,0038,0D80,0800,0000,1014  ;2464  D_K[.3]    ; Set the code to 11
            ;2465
            ;2466
            ;2467 =00 ID[39]_D,   ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1024  ;2468   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,106D  ;2469  J/RE.ENTRY   ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2470  RETURN2    ; Upper Controller configured
            ;2471       ; ...let the test know it
            ;2472 ;
            ;2473 ;
            ;2474 ; At this point the code is 11
            ;2475 ;
            ;2476 ;
            ;2477 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1018  ;2478  D_0    ; Clear the code
            ;2479 =00 ID[39]_D,   ; Save the code
U 1018, 0000,003D,E580,3C00,0000,11D4  ;2480   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2481       ; ...on the SBI
U 1019, 0000,003C,0180,0800,0000,106D  ;2482  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2483       ; ...attempt to configure the cntrllrs
U 101A, 0000,003E,0180,0800,0000,0001  ;2484  RETURN1    ; No more MS780-E/Hs found
U 101B, 0000,003C,0180,0800,0000,1020  ;2485  NOP
            ;2486
```

```
;2487  .PAGE
;2488  .TOC "   SELECT LOWER CONTROLLER"
;2489 ;*************************************************************
;2490 ;**
;2491 ;
;2492 ; Functional Description:
;2493 ; ----------------------
;2494 ;
;2495 ; This subroutine can be called to select the lower
;2496 ; Controller on a MS780-E/H.
;2497 ; If the Lower controller is misconfigured then a
;2498 ; RETURN1 will be made.  Otherwise a RETURN2
;2499 ;
;2500 ; Calling Constraint: =00
;2501 ; ------------------
;2502 ;
;2503 ;
;2504 ; Inputs:
;2505 ; ------
;2506 ;
;2507 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2508 ;
;2509 ; Outputs:
;2510 ; -------
;2511 ;
;2512 ; RC[0D] will have the address of CNFG Register C
;2513 ;        ( 2000x008 )
;2514 ;
;2515 ; Side Effects:
;2516 ; ------------
;2517 ;
;2518 ; Contents of the following registers will lost:
;2519 ;
;2520 ; D
;2521 ;
;2522 ; The Lower controller on the MS780-E/H will be configured
;2523 ; when the subroutine is exitted with a RETURN2
;2524 ;--
;2525 ;*************************************************************
;2526 =00
;2527 SELECT.LOWER:
;2528  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1020, 0818,0039,1980,0800,0000,116E  ;2529   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000,003E,0180,0800,0000,0001  ;2530  RETURN1    ; Lower Controller Misconfigured
U 1022, 0810,0038,0180,0970,0000,1023  ;2531  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2532  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1023, 0019,0016,0180,09E8,0000,0002  ;2533   RETURN2   ; Let caller know that the controller
;2534       ; ...was configured properly
```

```
;2535  .PAGE
;2536  .TOC "    SELECT UPPER CONTROLLER"
;2537 ;*************************************************************
;2538 ;++
;2539 ;
;2540 ; Functional Description:
;2541 ; ----------------------
;2542 ;
;2543 ; This subroutine can be called to select the upper
;2544 ; Controller on a MS780-E/H.
;2545 ; If the Upper controller is misconfigured then a
;2546 ; RETURN1 will be made.  Otherwise a RETURN2
;2547 ;
;2548 ; Calling Constraint: =00
;2549 ; ------------------
;2550 ;
;2551 ;
;2552 ; Inputs:
;2553 ; ------
;2554 ;
;2555 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2556 ;
;2557 ; Outputs:
;2558 ; -------
;2559 ;
;2560 ; RC[0D] will have the address of CNFG Register D
;2561 ;        ( 2000x010 )
;2562 ;
;2563 ; Side Effects:
;2564 ; ------------
;2565 ;
;2566 ; Contents of the following registers will lost:
;2567 ;
;2568 ; D
;2569 ;
;2570 ; The Upper controller on the MS780-E/H will be configured
;2571 ; when the subroutine is exitted with a RETURN2
;2572 ;--
;2573 ;*************************************************************
;2574 =00
;2575 SELECT.UPPER:
;2576  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,0980,0800,0000,116E  ;2577   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2578  RETURN1     ; Upper Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2579  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2580  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,8580,09E8,0000,0002  ;2581   RETURN2    ; Let caller know that the controller
;2582       ; ...was configured properly
;2583
```

```
            ;2584  .PAGE
            ;2585  .TOC "    SBI Unjam Routine"
            ;2586 ;++
            ;2587 ; FUNCTIONAL DESCRIPTION:
            ;2588 ;
            ;2589 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2590 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2591 ; and finally HOLD for the last 16 cycles.
            ;2592 ;
            ;2593 ; CALLING CONSTRAINT:
            ;2594 ;
            ;2595 ; =0
            ;2596 ;
            ;2597 ; SIDE EFFECTS:
            ;2598 ;
            ;2599 ; D Reg destroyed
            ;2600 ;--
            ;2601    ;
            ;2602    ; assert hold for 16 cycles
            ;2603    ;
            ;2604 CLRSBI:
            ;2605 SBI.UNJAM:
            ;2606  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 1184, 0818,0038,6580,8000,0010,108C  ;2607      clk.ubcc  ; set micro cc's for next cycle
            ;2608 =0
            ;2609 SBI.UNJAM.1:
            ;2610  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2611       clk.ubcc,z?, ; ...
U 108C, 0819,0100,0580,8000,0010,108C  ;2612      j/sbi.unjam.1 ; test for completion
            ;2613    ;
            ;2614    ; assert hold and unjam for 16 cycles
            ;2615    ;
            ;2616  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 108D, 0818,0038,6580,8800,0010,1094  ;2617      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2618 =0
            ;2619 SBI.UNJAM.2:
            ;2620  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2621       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1094, 0819,0100,0580,8800,0010,1094  ;2622      z?,j/sbi.unjam.2 ; ...
            ;2623    ;
            ;2624    ; assert hold for 16 cycles
            ;2625    ;
            ;2626  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1095, 0818,0038,6580,8000,0010,1096  ;2627      clk.ubcc  ; and set cc's for next cycle
            ;2628 =0
            ;2629 SBI.UNJAM.3:
            ;2630  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2631       clk.ubcc,z?, ; ...
U 1096, 0819,0100,0580,8000,0010,1096  ;2632      j/sbi.unjam.3 ; ...
U 1097, 0000,003E,0180,0800,0000,0001  ;2633  return1   ; exit
            ;2634
```

```
            ;2635  .PAGE
            ;2636  .TOC "   RESET SUBTEST NUMBER"
            ;2637 ;**
            ;2638 ; FUNCTIONAL DESCRIPTION:
            ;2639 ;
            ;2640 ; Since some  of the tests will have to be restarted when two
            ;2641 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2642 ; the subtest counter to zero ( only for thoes tests that have
            ;2643 ; more than one subtest). This subroutine may also be necessary
            ;2644 ; when a test is being loop on.
            ;2645 ;
            ;2646 ; CALLING CONSTRAINT:
            ;2647 ;
            ;2648 ; =0
            ;2649 ; SIDE EFFECTS:
            ;2650 ;
            ;2651 ; D is destroyed
            ;2652 ;
            ;2653 ;--
            ;2654 RESET.SUBTST:
            ;2655  RC[0F]_0,   ; Reset subtest number
U 1185, 0003,003E,0180,09F8,0000,0001  ;2656   RETURN1   ; ...and return
```

```
            ;2657  .PAGE
            ;2658  .TOC "   Initialize the SBI"
            ;2659 ;++
            ;2660 ; FUNCTIONAL DESCRIPTION:
            ;2661 ;
            ;2662 ; This routine performs the following functions:
            ;2663 ;
            ;2664 ;  Executes an SBI Unjam
            ;2665 ;  Clears the SBI Fault Latch
            ;2666 ;  Clears the SBI Error Register
            ;2667 ;
            ;2668 ; CALLING CONSTRAINT:
            ;2669 ;
            ;2670 ; =0
            ;2671 ;
            ;2672 ; SIDE EFFECTS:
            ;2673 ;
            ;2674 ; D & Q Register destroyed
            ;2675 ;--
            ;2676 =0
            ;2677 SBI.INIT:
U 1098, 0000,003D,0180,0800,0000,1184  ;2678  call[sbi.unjam]  ; Unjam the SBI
U 1099, 0818,0038,C180,0800,0000,1186  ;2679  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1186, 0801,0028,6580,3C00,0000,1187  ;2680  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1187, 0F00,003C,6D80,3C00,0000,1188  ;2681  id[fault]_d,d_0
U 1188, 0000,003C,6D80,3C00,0000,1189  ;2682  id[fault]_d
U 1189, 0000,003E,6580,3C00,0000,0001  ;2683  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2684
```

```
            ;2685  .PAGE
            ;2686  .TOC "   Enable Cache"
            ;2687 ;++
            ;2688 ; FUNCTIONAL DESCRIPTION:
            ;2689 ;
            ;2690 ; This routine enables cache group 0 by clearing the force
            ;2691 ; miss bit in the maintenance register.
            ;2692 ;
            ;2693 ; CALLING CONSTRAINT:
            ;2694 ;
            ;2695 ; =0
            ;2696 ;
            ;2697 ; SIDE EFFECTS:
            ;2698 ;
            ;2699 ; D, Q AND SC Registers destroyed
            ;2700 ;--
            ;2701 ENABLE.CACHE:
U 118A, 0000,003C,75F0,2C00,0000,118B  ;2702  q_id[maint]  ; Get current maintenance register
U 118B, 0000,003C,6580,0800,0084,718C  ;2703  sc_k[.10]  ; Setup to clear bit 16
U 118C, 0801,2034,0180,0800,0000,118D  ;2704  d_q.and.mask  ; Clear bit 16
U 118D, 0000,003E,7580,3C00,0000,0001  ;2705  id[maint]_d,return1 ; Rewrite register and return
            ;2706
```

```
            ;2707  .PAGE
            ;2708  .TOC "   Disable Cache Routine"
            ;2709 ;++
            ;2710 ; FUNCTIONAL DESCRIPTION:
            ;2711 ;
            ;2712 ; This routine disables cache hits by setting bits <16:15>
            ;2713 ; in the maintenance register.
            ;2714 ;
            ;2715 ; CALLING SEQUENCE:
            ;2716 ;
            ;2717 ; =0
            ;2718 ;
            ;2719 ; SIDE EFFECTS:
            ;2720 ;
            ;2721 ; D & Q Registers destroyed
            ;2722 ;--
            ;2723 DISABLE.CACHE:
U 118E, 0818,0038,4580,0800,0000,118F  ;2724  d_k[.8000]  ; ... and seed constant
U 118F, 0500,003C,0180,0800,0000,1190  ;2725  d_d.left  ; Generate final constant
U 1190, 0819,0030,4580,0800,0000,1191  ;2726  d_d.or.k[.8000]  ; ... = 00018000
U 1191, 0000,003E,7580,3C00,0000,0001  ;2727  id[maint]_d,return1 ; Disable cache and return
            ;2728
```

```
            ;2729  .PAGE
            ;2730  .TOC "   Check for Interrupt Routine"
            ;2731 ;++
            ;2732 ; FUNCTIONAL DESCRIPTION:
            ;2733 ;
            ;2734 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2735 ; If an interrupt is active, the following registers are setup:
            ;2736 ;  RC[8] - Gets the VECTOR register
            ;2737 ;  RC[7] - Gets the SBI ERROR register
            ;2738 ;  RC[6] - Gets the FAULT register
            ;2739 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2740 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2741 ;
            ;2742 ; CALLING CONSTRAINT:
            ;2743 ;
            ;2744 ; =00
            ;2745 ;
            ;2746 ; SIDE EFFECTS:
            ;2747 ;
            ;2748 ; D & Q are destroyed
            ;2749 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2750 ;--
            ;2751 CHECK_INTERRUPT:
            ;2752   ;
            ;2753   ; First, enable the fault and RDS/CRD interrupts
            ;2754   ;
U 1192, 0818,0038,4580,0800,0000,1193  ;2755  d_k[.8000]  ; Get data to set interrupt enable
            ;2756  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1193, 0500,003C,6580,3C00,0000,1194  ;2757       d_d.left
U 1194, 0100,003C,0180,0800,0000,1195  ;2758  d_d.left2  ; D = 40000
U 1195, 0000,003C,6D80,3C00,0000,1196  ;2759  id[fault]_d  ; Set fault interrupt enable
            ;2760  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1196, DF03,003C,0180,09A8,4000,1197  ;2761       rc[5]_0  ; and clear interrupt occurred flag
U 1197, 0000,003C,3D80,3C00,0000,1198  ;2762  id[psl]_d  ; Clear the PSL
U 1198, 0000,003C,6580,3C00,0000,1199  ;2763  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1199, 0000,003C,6D80,3C00,0000,119A  ;2764  id[fault]_d  ; Clear FAULT Interrupt Enable
U 119A, 0000,0E3C,0180,0800,0000,1004  ;2765  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2766 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2767  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2768  return1   ; No interrupt
            ;2769 =111
            ;2770 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,119B  ;2771  q_id[vector]  ; Get ID Vector Register
U 119B, 0001,203C,65F0,2DC0,0000,119C  ;2772  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 119C, 0001,203C,6DF0,2DB8,0000,119D  ;2773  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 119D, 0001,203C,0180,09B0,0000,119E  ;2774  rc[6]_q   ; Save falut reg
U 119E, 0018,003A,0580,09A8,0000,0002  ;2775  rc[5]_k[.1],return2 ; Set error flag and return
            ;2776
```

```
            ;2777  .PAGE
            ;2778  .TOC "   Set Trap Mask"
            ;2779 ;**
            ;2780 ; FUNCTIONAL DESCRIPTION:
            ;2781 ;
            ;2782 ; This routine is used to set a bit in the Micro Trap Mask
            ;2783 ; R[0F].
            ;2784 ;
            ;2785 ; CALLING CONSTRAINT:
            ;2786 ;
            ;2787 ; =0
            ;2788 ;
            ;2789 ; INPUTS:
            ;2790 ;
            ;2791 ; SC - Contains bit number to set.
            ;2792 ;
            ;2793 ; OUTPUTS:
            ;2794 ;
            ;2795 ; rc[0E] - Contains the current trap mask
            ;2796 ;
            ;2797 ; SIDE EFFECTS:
            ;2798 ;
            ;2799 ; d regsiter is destroyed
            ;2800 ;--
            ;2801 SET_TRAP_MASK:
U 119F, 0800.003C.0180.0A78.0000.11A0  ;2802  d_r[0f]
U 11A0, 0801.001C.0180.0AF8.0000.11A1  ;2803  r[0f]_d.ornot.mask,d_alu ; Set mask
U 11A1, 0001.003E.0180.0AF0.0000.0001  ;2804  r[0E]_d,return1    ; Save mask and return
            ;2805
```

```
            ;2806  .PAGE
            ;2807  .TOC "    FIND MS780-E MEMORY"
            ;2808 ;++
            ;2809 ; FUNCTIONAL DESCRIPTION:
            ;2810 ;
            ;2811 ;          The diagnostic is designed to handle up to 4 (four)
            ;2812 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2813 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2814 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2815 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2816 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2817 ;     first MS780-E found will have its starting address set to 0
            ;2818 ;     (zero).
            ;2819 ;     All the other MS780-E/H's found will have their starting address
            ;2820 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2821 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2822 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2823 ;     current MS780-E unit under test.
            ;2824 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2825 ;     set to maximum.
            ;2826 ;
            ;2827 ; CALLING CONSTRAINT:
            ;2828 ;
            ;2829 ; =00
            ;2830 ;
            ;2831 ; OUTPUTS:
            ;2832 ;
            ;2833 ; rc[0e] - Contains address of Configuration Register
            ;2834 ; r[0]  - Contains TR level
            ;2835 ;
            ;2836 ; SIDE EFFECTS:
            ;2837 ;
            ;2838 ; D register is destroyed.
            ;2839 ;--
            ;2840 =0
            ;2841 FIND.MS780E:
U 109A. 0000,003D,0180,0800,0000,1098  ;2842  call[sbi.init]  ; Make sure SBI is enabled
U 109B. 0003,003C,0180,0988,0000,11A2  ;2843  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11A2. 0818,0038,1980,0800,0000,11A3  ;2844  d_k[zero]  ; Clear MS780-E/H counter
U 11A3. 0000,003C,F980,3C00,0000,11A4  ;2845  id[3e]_d  ; ...
U 11A4. 0018,0038,0580,0A80,0000,11A5  ;2846  r[0]_k[.1]  ; Init TR counter
U 11A5. 0F03,003C,0180,09F0,0000,109C  ;2847  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2848      ; ...CNFG A Reg
            ;2849 =0
            ;2850 FIND.MEM.LOOP:
U 109C. 0800,003D,0180,0A00,0000,11CB  ;2851  d_r[0],call[generate.io]; Generate address of TR level
U 109D. 0001,003C,0180,09F0,0200,10A0  ;2852  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10A0. 0000,003D,8980,0800,0084,719F  ;2853 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2854  d[long]_cache.p, ; Read the specified tr level
U 10A1. 0000,003C,0180,C970,0000,11A6  ;2855    ic_rc[0e]  ; ...
U 11A6. 0000,003C,0180,0A78,0010,11A7  ;2856  alu_r[0f],clk.ubcc ; Check for no response
U 11A7. 0001,013C,0180,0880,0082,10A2  ;2857  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10A2. 0000,003C,0180,0800,0000,11A8  ;2858 =0 j/check.adpt.code ; Check for a memory
U 10A3. 0000,003C,0180,0800,0000,11C7  ;2859  j/find.mem.lp1  ; Try next tr
            ;2860 CHECK.ADPT.CODE:
```

```
U 11A8, 0000,003C,1D80,0800,1404,71A9  ;2861  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11A9, 0000,163C,0180,0800,0000,1028  ;2862  state7-4?  ; Branch on the adapter type
U 1028, 0000,003C,8980,0800,0084,71AA  ;2863 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1029, 0000,003C,8980,0800,0084,71AB  ;2864  j/ms780.c,sc_k[.d] ; MS780-C
U 102A, 0000,003C,0180,0800,0000,11C7  ;2865  j/find.mem.lp1  ; Not a memory
U 102B, 0000,003C,0180,0800,0000,11C7  ;2866  j/find.mem.lp1  ; Not a memory
U 102C, 0000,003C,6580,0800,0084,71B0  ;2867  j/ma780.max,sc_k[.10] ; MA780
U 102D, 0000,003C,0180,0800,0000,11C7  ;2868  j/find.mem.lp1  ; Not a memory
U 102E, 0010,0038,0180,09F0,0000,11B4  ;2869  rc[0e]_lc,j/found.ms780e ; MS780-E
U 102F, 0010,0038,0180,09F0,0000,11B4  ;2870  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2871   ;
            ;2872   ; Found an MS780-A or -C. Set the starting address
            ;2873   ; to maximum.
            ;2874   ;
U 11AA, 0818,0038,5D80,0800,0000,11AC  ;2875 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11AB, 0818,0038,0580,0800,0000,11AC  ;2876 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2877 MS780.COMMON:
U 11AC, 0819,0030,7180,0800,0000,11AD  ;2878  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11AD, 0000,003C,01F8,0803,0080,D1AE  ;2879  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11AE, 0D00,003C,0180,0800,0000,11AF  ;2880  d_dal.sc  ; Put data in bits<29:14>
            ;2881  cache.p_d[long], ; Set the starting address to maximum
U 11AF, 0000,003C,0180,A800,0000,11C7  ;2882       j/find.mem.lp1 ; and continue TR scan
            ;2883   ;
            ;2884   ; Found an MA780. Set the starting address to maximum
            ;2885   ;
            ;2886 MA780.MAX:
U 11B0, 0818,0038,39F8,0803,0000,11B1  ;2887  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11B1, 0D00,003C,0180,0803,0000,11B2  ;2888  d_dal.sc,va_va+4 ; Put in proper position
U 11B2, 0000,003C,0180,0803,0000,11B3  ;2889  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2890  cache.p_d[long], ; Set starting address to maximum
U 11B3, 0000,003C,0180,A800,0000,11C7  ;2891       j/find.mem.lp1
            ;2892 ;
            ;2893 ; Found an MS780E
            ;2894 ;
            ;2895 FOUND.MS780E:
U 11B4, 0000,003C,F9F0,2C00,0000,11B5  ;2896  q_id[3e]  ; Set up to see how many MS780-E's
            ;2897  alu_q.xor.k[.3], ; Are there Maximum units?
U 11B5, 0019,2020,0D80,0800,0010,11B6  ;2898   clk.ubcc ; Check condition codes
U 11B6, 0000,013C,0180,0800,0000,10A4  ;2899  z?   ; Are we at maximum units for system
U 10A4, 0000,003C,0180,0800,0000,11B7  ;2900 =0 J/ms780e.tst  ; No go find how many units ther are
U 10A5, 0818,0038,C180,0800,0000,10A8  ;2901  d_k[.ffff]  ; Yes set address to maximum
U 10A8, 0000,003D,0180,0800,0000,11CF  ;2902 =0 call[set.start.addr] ; .....
U 10A9, 0000,003C,0180,0800,0000,11C7  ;2903  j/find.mem.lp1  ; Go check to see if we are done
            ;2904 ;
            ;2905 MS780E.TST:
            ;2906  alu_rc[01],  ; Check "Found MS780E Flag"
U 11B7, 0010,0038,0180,0908,0010,11B8  ;2907   clk.ubcc ; Check condition codes
U 11B8, 0810,0138,0180,0970,0000,10AA  ;2908  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2909 =0 j/not.first.ms780e, ; No
U 10AA, 0818,0038,C180,0800,0000,10B0  ;2910   d_k[.ffff] ; Set starting address
U 10AB, 0001,003C,0180,0988,0000,11B9  ;2911  rc[01]_d  ; Yes put base address in rc[01]
U 11B9, 0818,0038,7980,0800,0000,11BA  ;2912  d_k[.30]  ; Set up SC to point to first unit
U 11BA, 0019,0014,F580,0800,0082,11BB  ;2913  alu_d+k[.a],sc_alu ; ...
U 11BB, 0810,0038,0180,0908,0000,11BC  ;2914  d_rc[01]  ; Put base address of unit in D
U 11BC, 0000,003C,0180,3400,0000,11BD  ;2915  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11BD. 0F00,003C,0180,0800,0000,10AC  ;2916  d_0    ; Prepare to set starting address to Zero
U 10AC. 0000,003D,0180,0800,0000,11CF  ;2917 =0 call[set.start.addr] ; Go do it
            ;2918  j/find.mem.lp1.   ; Check to see if we are done
U 10AD. 0003,003C,0180,09E8,0000,11C7  ;2919   rc[0d]_0 ; ....
            ;2920 =0
            ;2921 NOT.FIRST.MS780E:
U 10B0. 0000,003D,0180,0800,0000,11CF  ;2922  call[set.start.addr] ; Go set starting address to maximum
U 10B1. 0000,003C,F9F0,2C00,0000,11BE  ;2923  q_id[3e]  ; Get the number of MS780-Es found
U 11BE. 0819,2014,0580,0800,0000,11BF  ;2924  d_q+k[.1]  ; Increment this counter
U 11BF. 0000,003C,F980,3C00,0000,11C0  ;2925  id[3e]_d  ; Save the counter
U 11C0. 0018,0038,11C0,0800,0000,11C1  ;2926  q_k[.4]   ; Prepare to form pointer to the ID registers
            ;2927      ; ...were the I/O base addresses will be stored
U 11C1. 081D,2000,7980,0800,0000,11C2  ;2928  d_q-d,k[.30]  ; ... adjust pointer
U 11C2. 0819,0014,7980,0800,0000,11C3  ;2929  d_d+k[.30]  ; Point the SC to the ID register
U 11C3. 0000,003C,F580,0800,0000,11C4  ;2930  k[.a]   ; ...to receive I/O base address
U 11C4. 0019,0014,F580,0800,0082,11C5  ;2931  alu_d+k[.a],sc_alu ; ...
U 11C5. 0810,0038,0180,0970,0000,11C6  ;2932  d_rc[0e]  ; Get base address to d
U 11C6. 0000,003C,0180,3400,0000,11C7  ;2933  id(sc)_d  ; Move base address to ID pointed to by SC
            ;2934   ;
            ;2935   ; Try next tr
            ;2936   ;
            ;2937 FIND.MEM.LP1:
U 11C7. 0818,0014,0580,0A80,0000,11C8  ;2938  r[0]_ia+k[.1],d_alu ; Increment TR level
            ;2939  alu_d.and.k[.f],
U 11C8. 0019,0034,6180,0800,0010,11C9  ;2940      clk.ubcc  ; Check if all done
U 11C9. 0000,013C,0180,C800,0000,10B2  ;2941  z?    ; Branch if yes
U 10B2. 0000,003C,0180,0800,0000,109C  ;2942 =0 j/find.mem.loop  ; Try next TR
U 10B3. 0810,0038,0180,0908,0010,11CA  ;2943  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11CA. 0000,013C,0180,0800,0000,10B4  ;2944  z?    ; Check condition codes
U 10B4. 0001,003E,0180,09F0,0000,0002  ;2945 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10B5. 0000,003E,0180,0800,0000,0001  ;2946  return1    ; No MS780E found
            ;2947
```

```
            ;2948  .PAGE
            ;2949  .TOC "   Generate IO Address"
            ;2950 ;++
            ;2951 ; FUNCTIONAL DESCRIPTION:
            ;2952 ;
            ;2953 ; This routine is used to generate an SBI Configuration Register
            ;2954 ; address from a TR level.
            ;2955 ;
            ;2956 ; CALLING CONSTRAINT:
            ;2957 ;
            ;2958 ; =0
            ;2959 ;
            ;2960 ; INPUTS:
            ;2961 ;
            ;2962 ; D - Contains TR level
            ;2963 ;
            ;2964 ; OUTPUTS:
            ;2965 ;
            ;2966 ; D - Contains SBI IO address
            ;2967 ;--
            ;2968 GENERATE.IO:
U 11CB, 0000,003C,89F8,0800,0084,71CC  ;2969  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11CC, 0D00,003C,8D80,0800,0084,71CD  ;2970  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11CD, 0000,003C,0980,0800,0084,B1CE  ;2971  sc_sc-k[.2]  ; insert bit 29
U 11CE, 0801,001E,0180,0800,0000,0001  ;2972  d_d.ornot.mask,return1 ; and return
            ;2973
```

```
            ;2974  .PAGE
            ;2975  .TOC "    Set Starting Address"
            ;2976 ;++
            ;2977 ; FUNCTIONAL DESCRIPTION:
            ;2978 ;
            ;2979 ; This routine is used to set the starting address of the
            ;2980 ; memory to the specified value.
            ;2981 ;
            ;2982 ; CALLING CONSTRAINT:
            ;2983 ;
            ;2984 ; =0
            ;2985 ;
            ;2986 ; INPUTS:
            ;2987 ;
            ;2988 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2989 ;      (B Register Image divided by 2**16)
            ;2990 ; rc[0e] - Contains Address of Register A
            ;2991 ;
            ;2992 ; OUTPUTS:
            ;2993 ;
            ;2994 ; d - Contains aligned starting address (B Register Image)
            ;2995 ;
            ;2996 ; SIDE EFFECTS:
            ;2997 ;
            ;2998 ; d,q,va, and sc are destroyed
            ;2999 ;--
            ;3000 SET.START.ADDR:
U 11CF, 0010,0038,6580,0970,0284,71D0  ;3001  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11D0, 0000,003C,01F8,0803,0000,11D1  ;3002  va_va+4,q_0  ; Set address to Register B
            ;3003  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11D1, 0D00,003C,0980,0800,0084,B1D2  ;3004       sc_sc-k[.2]  ; Setup to insert bit 14
U 11D2, 0801,001C,0180,0800,0000,11D3  ;3005  d_d.ornot.mask  ; Insert Write Enable bit
U 11D3, 0000,003E,0180,A800,0000,0001  ;3006  cache.p_d[long],return1 ; Set the starting address and return
            ;3007
```

```
            ;3008  .PAGE
            ;3009  .TOC "   ENABLE NEXT MS780-E"
            ;3010 ;**
            ;3011 ; FUNCTIONAL DESCRIPTION:
            ;3012 ;
            ;3013 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3014 ;    They will be tested  separately, according to the TR level at which
            ;3015 ;    they are located,  starting with the one  at the lowest level first.
            ;3016 ;          When  a test has finished  with the first unit, it will have to call
            ;3017 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3018 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3019 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3020 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3021 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3022 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3023 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3024 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3025 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3026 ;
            ;3027 ; CALLING CONSTRAINT:
            ;3028 ;
            ;3029 ;  =00
            ;3030 ;
            ;3031 ;--
            ;3032 ENABLE.NEXT.MS780E:
U 11D4, 0000,003C,F9F0,2C00,0000,11D5  ;3033  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3034  ALU_Q,   ; Check to see if it is zero
U 11D5, 0001,203C,0180,0800,0010,11D6  ;3035      CLK.UBCC  ; Check Condition Codes
U 11D6, 0000,013C,0180,0800,0000,10B6  ;3036  Z?   ; ...for zero
U 10B6, 0000,003C,0180,0B00,0000,11D7  ;3037 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10B7, 0000,003E,0180,0800,0000,0002  ;3038  RETURN2   ; Zero No more units to test
            ;3039 ENABLE.NEXT:
U 11D7, 0818,0038,C180,0B00,0000,10B8  ;3040  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10B8, 0000,003D,0180,0B00,0000,11CF  ;3041 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10B9, 0818,0038,7980,0800,0000,11D8  ;3042  D_K[.30]  ; Prepare to point to the ID register holding
            ;3043      ; ...the I/O Base address of the next MS780-E
U 11D8, 0819,0014,1180,0800,0000,11D9  ;3044  D_D+K[.4]  ; ...
U 11D9, 0000,003C,F580,0800,0000,11DA  ;3045  K[.A]   ; ...
U 11DA, 0819,0014,F580,0800,0000,11DB  ;3046  D_D+K[.A]  ; ...
U 11DB, 0000,003C,F9F0,2C00,0000,11DC  ;3047  Q_ID[3E]  ; Get MS780-E Counter
            ;3048  D_D-Q,   ; D now holds the pointer to the ID register
U 11DC, 081D,0000,0180,0800,0082,11DD  ;3049   SC_ALU  ; ...
U 11DD, 0000,003C,01F0,2400,0000,11DE  ;3050  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11DE, 0001,203C,0180,09F0,0000,11DF  ;3051  RC[0E]_Q  ; Save it also in RC[0E]
U 11DF, 0F03,003C,0180,09E8,0000,10BA  ;3052  RC[0D]_0,D_0  ; Set starting address to zero
U 10BA, 0000,003D,0180,0800,0000,11CF  ;3053 =0 CALL[SET.START.ADDR] ; ....
U 10BB, 0F00,003C,F9F0,2C00,0000,11E0  ;3054  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11E0, 081D,2008,0180,0800,0000,11E1  ;3055  D_Q-D-1   ; Subtract 1 from total
U 11E1, 0000,003C,F980,3C00,0000,10BC  ;3056  ID[3E]_D  ; Adjust the pointer
U 10BC, 0000,003D,0180,0800,0000,1185  ;3057 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10BD, 0000,003E,0180,0800,0000,0001  ;3058  RETURN1   ; Tell caller another unit was found
            ;3059
```

```
            ;3060  .PAGE
            ;3061  .TOC "   GET MEMORY SIZE"
            ;3062 ;*************************************************************
            ;3063 ;++
            ;3064 ;
            ;3065 ; Functional Description:
            ;3066 ; ----------------------
            ;3067 ;
            ;3068 ; This subroutine can be used to get the size of the memory.
            ;3069 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3070 ; increments.
            ;3071 ;
            ;3072 ; Calling Constraint: =0
            ;3073 ; ------------------
            ;3074 ;
            ;3075 ;
            ;3076 ; Inputs:
            ;3077 ; ------
            ;3078 ;
            ;3079 ; RC[0E] Must have the Address of CNFG Register A
            ;3080 ;
            ;3081 ; Outputs:
            ;3082 ; -------
            ;3083 ;
            ;3084 ; Register D will contain upon exit the size of the
            ;3085 ; memory aligned on Mega-byte boundary.
            ;3086 ;
            ;3087 ; Side Effects:
            ;3088 ; ------------
            ;3089 ;
            ;3090 ; The contents of the following registers will be lost:
            ;3091 ;
            ;3092 ; D, SC, Q
            ;3093 ;
            ;3094 ;--
            ;3095 ;*************************************************************
            ;3096 GET.MEMORY.SIZE:
U 11E2, 0010,0038,0180,0970,0200,11E3  ;3097  VA_RC[0E]   ; Point to CNFG Reg A
U 11E3, 0000,003C,0180,C800,0000,11E4  ;3098  D[LONG]_CACHE.P   ; Read the register
U 11E4, 001B,0000,D980,0800,0082,11E5  ;3099  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 11E5, 0D00,003C,0180,0800,0000,11E6  ;3100  D_DAL.SC   ; Shift bits <14:9> to <5:0>
U 11E6, 0819,0034,5580,0800,0000,1030  ;3101  D_D.AND.K[.3F]   ; Mask unwanted bits and clear Reg Q
            ;3102 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1030, 0000,003D,01E0,0800,0000,1178  ;3103   Q_D   ; Save Memory size bits in Q
U 1031, 0000,003D,0D80,0800,0084,704C  ;3104  ERROR1[.3]   ; RET1 Log an Error. Mixed or Missing
            ;3105       ; ...arrays on the controller
            ;3106 ;
            ;3107 ; The duplication of the next two return status is necessary so the returning
            ;3108 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3109 ; modifications.
            ;3110 ;
U 1032, 0819,2014,0580,0800,0000,1033  ;3111  D_Q+K[.1]   ; RET2 Increment memory size by one
            ;3112       ; ...(found 1 Meg arrays)
U 1033, 0819,2014,0580,0800,0000,11E7  ;3113  D_Q+K[.1]   ; RET3 Increment memory size by 1
            ;3114       ; ...(found 4 Meg arrays)
```

```
U 11E7, 0000,003C,21F8,0800,0084,71E8  ;3115  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
          ;3116  D_DAL.SC,   ; Shift bits <5:0> to <25:20>
U 11E8, 0D00,003E,0180,0800,0000,0001  ;3117   RETURN1   ; Return with memory size in Register D
          ;3118
```

```
            ;3119  .PAGE
            ;3120  .TOC "   ERDAT ROUTINE"
            ;3121 ;
            ;3122 ;////////////////////////////////////////////////////////////////////////////
            ;3123 ;++
            ;3124 ;       THIS SUBROUTINE GETS THE ERROR DATA FROM ID[25-2X]
            ;3125 ; AND PUTS THAT DATA IN RC[0E]-RC[0X].
            ;3126 ;
            ;3127 ; (X DEPENDS ON HOW MANY ERRORS OCCURRED; MAXIMUM OF 6 DECIMAL)
            ;3128 ;
            ;3129 ; IT IS ASSUMMED THAT THE NUMBER OF ERRORS IS CONTAINED
            ;3130 ; IN THE STATE REGISTER.
            ;3131 ; (ALL F'S IS THE FLAG FOR END OF ERROR DATA).
            ;3132 ;--
            ;3133 ;////////////////////////////////////////////////////////////////////////////
            ;3134 ;
U 11E9. 0000.003C.8580.0800.0184.71EA  ;3135 ERDAT: FE&SC_K[.C]    ; INIT RC ADRS TO C
U 11EA. 0000.003C.B980.0800.0084.91EB  ;3136  SC_SC+K[.19]    ; INIT ID ADRS TO 25 (HEX)
U 11EB. 0000.003C.01F0.2400.0000.11EC  ;3137 ERDLP: Q_ID(SC)
U 11EC. 0000.003C.0180.0800.0181.D1ED  ;3138  FE_SC+1,SC_FE    ; INCR ID ADRS,SC_RC ADRS
U 11ED. 0001.203C.0180.0838.0000.11EE  ;3139  RC(SC)_ALU,ALU_Q
U 11EE. 0000.003C.0580.0800.0185.B1EF  ;3140  SC_FE,FE_SC-K[.1]   ; SC_ID ADRS,FE_DECR'D RC ADRS
U 11EF. 0000.003C.0580.0800.1414.B1F0  ;3141  STATE_STATE-K[.1],CLK.UBCC ; DECR ERROR COUNTER
U 11F0. 0000.123C.0180.0800.0000.103B  ;3142  EALU.Z?     ; DONE LOOP ERROR # TIMES?
U 103B. 0000.003C.0180.0800.0000.11EB  ;3143 =1011 J/ERDLP     ; NOPE
U 103F. 0000.003C.0180.0800.0081.11F1  ;3144  SC_FE     ; PUT END FLAG IN RC[X]
            ;3145  RETURN1,    ; ...AND RETURN TO ERROR CALL.
U 11F1. 001B.0002.0580.0838.0000.0001  ;3146   RC(SC)_ALU,ALU_0-K[.1]
            ;3147
```

```
            ;3148  .PAGE
            ;3149  .TOC "   Set Diagnostic Mode Routine"
            ;3150 ;++
            ;3151 ; FUNCTIONAL DESCRIPTION:
            ;3152 ;
            ;3153 ; This routine is used to set the specified diagnostic mode
            ;3154 ; in Register B. Other bits in the register remain unchanged.
            ;3155 ;
            ;3156 ; CALLING CONSTRAINT:
            ;3157 ;
            ;3158 ; =0
            ;3159 ;
            ;3160 ; INPUTS:
            ;3161 ;
            ;3162 ; d - Contains the mode to set in bits<1:0>
            ;3163 ; rc[0e] - Contains base address of controller
            ;3164 ;
            ;3165 ; SIDE EFFECTS:
            ;3166 ;
            ;3167 ; d,va,sc are destroyed
            ;3168 ;--
            ;3169 SET_DIAG_MODE:
U 11F2, 0010,0038,D9F8,0970,0284,71F3  ;3170  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11F3, 0D00,003C,0180,0803,0000,11F4  ;3171  va_va+4,d_dal.sc ; Shift specified mode into position
U 11F4, 0000,003C,01E0,0800,0000,11F5  ;3172  q_d    ; Save the diagnostic mode bits
U 11F5, 0000,003C,0180,C800,0000,11F6  ;3173  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11F6, 081D,0030,0180,0800,0000,11F7  ;3174  d_d.or.q  ; Set the diagnostic mode bits
U 11F7, 0000,003E,0180,A800,0000,0001  ;3175  cache.p_d[long],return1 ; Set the specified mode and return
            ;3176
```

```
;3177  .PAGE
;3178  .TOC "    Check for Odd or Even TR"
;3179 ;*************************************************************
;3180 ;++
;3181 ;
;3182 ; Functional Description:
;3183 ; ----------------------
;3184 ;
;3185 ;
;3186 ;      This subroutine will  determine  base  on  the  contents  of
;3187 ; RC[0E]  (which  contains  the  I/O  base register address for the
;3188 ; MS780-E/H currently under test), whether the memory is at Even or
;3189 ; Odd TR level.
;3190 ;
;3191 ;
;3192 ; Calling Constraint: =00
;3193 ; ------------------
;3194 ;
;3195 ;
;3196 ; Inputs:
;3197 ; ------
;3198 ;
;3199 ; RC[0E] must hold the I/O base address of the memory under test
;3200 ;
;3201 ;
;3202 ; Outputs:
;3203 ; -------
;3204 ;
;3205 ; There are no specific outputs for this routine.
;3206 ; The only output will be the RETURN mode:
;3207 ;
;3208 ;  RETURN1 for EVEN TR
;3209 ;
;3210 ;  RETURN2 for ODD TR
;3211 ;
;3212 ;
;3213 ; Side Effects:
;3214 ; ------------
;3215 ;
;3216 ; The contents of the following registers will be lost
;3217 ;
;3218 ; D, SC
;3219 ;
;3220 ;--
;3221 ;*************************************************************
;3222 ODD.EVEN.TR:
U 11F8, 081B.001C.8580.0800.0000.11F9  ;3223  D_NOT.K[.C]    ; Load D register with negative x12
U 11F9, 0001.003C.0180.0800.0082.11FA  ;3224  SC_D     ; Prepare SC for right shift of x12
U 11FA, 0810.0038.0180.0970.0000.11FB  ;3225  D_RC[0E]    ; Get the I/O base address
U 11FB, 0000.003C.01F8.0800.0000.11FC  ;3226  Q_0     ; Clear Q Reg
U 11FC, 0D00.003C.0180.0800.0000.11FD  ;3227  D_DAL.SC    ; Shift D right 13 bits
U 11FD, 0000.193C.0180.0800.0000.106E  ;3228  D1-0?     ; Is bit 0 of D set?
U 106E, 0000.003E.0180.0800.0000.0001  ;3229 =1110 RETURN1     ; EVEN TR
U 106F, 0000.003E.0180.0800.0000.0002  ;3230  RETURN2     ; ODD TR
;3231
```

```
;3232
```

```
;3233  .PAGE
;3234  .TOC "    RAS ADDRES GENREATOR"
;3235 ;*************************************************************
;3236 ;**
;3237 ;
;3238 ; Functional Description:
;3239 ; ----------------------
;3240 ;
;3241 ; This subtest will be used by the RAS/CAS test to generate a
;3242 ; RAS address  based on  whether the  arrays have 64K or 256K
;3243 ; chips on and a Pattern Counter passed by the callin routine.
;3244 ;
;3245 ; Calling Constraint: =0
;3246 ; ------------------
;3247 ;
;3248 ;
;3249 ; Inputs:
;3250 ; ------
;3251 ;
;3252 ; RC[04] RAS Counter (Pattern) on which the RAS address is based
;3253 ;
;3254 ; RC[0C] FLAG indicating the type of chips used
;3255 ;
;3256 ;  0 = 64K chips
;3257 ;
;3258 ;  1 = 256K chips
;325° ;
;3260 ;
;3261 ; Outputs:
;3262 ; -------
;3263 ;
;3264 ; Upon exit the D register will hold the RAS address
;3265 ;
;3266 ;
;3267 ; Side Effects:
;3268 ; ------------
;3269 ;
;3270 ; The contents of the following registers will be lost:
;3271 ;
;3272 ; D, SC
;3273 ;
;3274 ;--
;3275 ;*************************************************************
;3276 RAS.ADD.GEN:
U 11FE, 0810,0038,0180,0920,0000,11FF  ;3277  D_RC[4]    ; Get the RAS Pattern Counter
        ;3278  D_D.AND.K[.FF],   ; Mask unwanted bits
U 11FF, 0819,0034,49F8,0800,0000,1200  ;3279   Q_0    ; ...clear the Q register
U 1200, 0000,003C,1180,0800,0084,7201  ;3280  SC_K[.4]   ; Prepare to shift the address
        ;3281       ; ...4 bits to the left
U 1201, 0D00,003C,0180,0800,0000,1202  ;3282  D_DAL.SC    ; Shift
U 1202, 0010,0038,01C0,0920,0000,1203  ;3283  Q_RC[4]    ; Get the RAS Pattern Counter
U 1203, 0000,003C,0180,0800,0084,7204  ;3284  SC_K[.8]    ; Point MASK to bit 8
        ;3285  ALU_Q.ANDNOT.MASK,  ; See if bit 8 is set in the RAS
U 1204, 0001,2024,0180,0800,0010,1205  ;3286   CLK.UBCC   ; ...counter
U 1205, 0000,013C,2180,0800,0084,70BE  ;3287  Z?,SC_K[.14]    ;
```

```
U 10BE, 0801,001C,0180,0800,0000,10BF  ;3288 =0 D_D.ORNOT.MASK   ; Bit was set, so set bit 20
          ;3289       ; ...in the final RAS address
U 10BF, 0000,003E,0180,0800,0000,0001  ;3290  RETURN1    ; Bit was not set
          ;3291       ; ...return to caller
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1207
ESKAR4A.MCR                   MICRO2  1P(00)    26-JUL-85  17:26:47                    Page   9

            ;3292  .PAGE
            ;3293  .TOC "    CAS ADDRESS GENERATOR"
            ;3294 ;*************************************************************
            ;3295 ;**
            ;3296 ;
            ;3297 ; Functional Description:
            ;3298 ; ----------------------
            ;3299 ;
            ;3300 ; This subtest will be used by the RAS/CAS test to generate
            ;3301 ; a  CAS address, based on whether  the arrays have  64K or
            ;3302 ; 256K chip and on a Pattern Counter.
            ;3303 ;
            ;3304 ; Calling Constraint: =0
            ;3305 ; ------------------
            ;3306 ;
            ;3307 ;
            ;3308 ; Inputs:
            ;3309 ; ------
            ;3310 ;
            ;3311 ; RC[04] Pattern used for CAS Address generation
            ;3312 ;
            ;3313 ; RC[0C] Flag indicating the type of chips being used
            ;3314 ;
            ;3315 ;  0 = 64K chips
            ;3316 ;
            ;3317 ;  1 = 256K chips
            ;3318 ;
            ;3319 ;
            ;3320 ; Outputs:
            ;3321 ; -------
            ;3322 ;
            ;3323 ; Upon exit Register D will hold the CAS address
            ;3324 ;
            ;3325 ;
            ;3326 ; Side Effects:
            ;3327 ; ------------
            ;3328 ;
            ;3329 ; The contents of the following registers will be lost:
            ;3330 ;
            ;3331 ; D, Q, SC
            ;3332 ;
            ;3333 ;--
            ;3334 ;*************************************************************
            ;3335 CAS.ADD.GEN:
U 1206, 0810.0038.0180.0920.0000,1207  ;3336  D_RC[4]     ; Get the CAS Pattern Counter
U 1207, 0819.0034.4980.0800.0000,1208  ;3337  D_D.AND.K[.FF]    ; Mask unwanted bits
            ;3338  SC_K[.C],    ; Prepare to shift the address
U 1208, 0000.003C.85F8.0800.0084,7209  ;3339    Q_0    ; ...bits in position
U 1209, 0D00.003C.0180.0800.0000,120A  ;3340  D_DAL.SC    ; Shift
U 120A, 0010.0038.01C0.0920.0000,120B  ;3341  Q_RC[4]     ; Get the Pattern counter
U 120B, 0000.003C.0180.0800.0084,720C  ;3342  SC_K[.8]    ; Point sc to bit 8
            ;3343  ALU_Q.ANDNOT.MASK,  ; Is bit 8 set in the CAS counter?
U 120C, 0001.2024.0180.0800.0010,120D  ;3344    CLK.UBCC   ; ...
U 120D, 0000.013C.2180.0800.0084,70C0  ;3345  Z?.SC_K[.14]    ; SC = d20
            ;3346 =0 SC_SC+1,    ; Point SC to bit 21
```

```
U 10C0. 0000.003C.0180.0800.0080.D20E  ;3347   J/CAS.SET.BIT.21 ; ...and set it in the final address
U 10C1. 0000.003E.0180.0800.0000.0001  ;3348  RETURN1    ; Bit was NOT set so return to caller
            ;3349 CAS.SET.BIT.21:
            ;3350  D_D.ORNOT.MASK,   ; Bit 8 was set in the CAS counter
            ;3351       ; ...so set bit 21 in the CAS address
U 120E. 0801.001E.0180.0800.0000.0001  ;3352   RETURN1   ; Return to caller
            ;3353
            ;3354
            ;3355
```

```
;3356  .PAGE
;3357  .TOC "   REED MULLER ENCODING ROUTINE"
;3358 ;*****************************************************************
;3359 ;**
;3360 ;
;3361 ; Functional Description:
;3362 ; ----------------------
;3363 ;
;3364 ;      This subroutine is used to apply the Reed-Muller
;3365 ; encoding  algorithm to a 16 bit  vector with an error
;3366 ; handling capability of up to 3 bits per 32 bit vector.
;3367 ;
;3368 ;
;3369 ; NOTE: The DESIGN SPEC for the MS780-E/H diagnostic
;3370 ;  includes a theoretical discussion of the
;3371 ;  Reed-Muller codes as well as the algorithm
;3372 ;  for the following subroutines:
;3373 ;
;3374 ;  ENCODE
;3375 ;
;3376 ;  DECODE
;3377 ;
;3378 ;  FIRST.ORDER
;3379 ;
;3380 ;  SECOND.ORDER
;3381 ;
;3382 ;
;3383 ; Calling Constraint: =0
;3384 ; ------------------
;3385 ;
;3386 ;
;3387 ; Inputs:
;3388 ; ------
;3389 ;
;3390 ; RC[4] must contain the 16 bit vector that should
;3391 ; be encoded.
;3392 ;
;3393 ; Outputs:
;3394 ; -------
;3395 ;
;3396 ; RC[1] will contain upon exit the encoded vector
;3397 ;
;3398 ;
;3399 ; Side Effects:
;3400 ; ------------
;3401 ;
;3402 ; The contents of the following registers will be lost:
;3403 ;
;3404 ; R[0] through R[0B], RC[1], T1 through T6
;3405 ;
;3406 ;--
;3407 ;*****************************************************************
;3408 ENCODE:
U 120F, 0018,0038,0580,0AD0,0000,1210  ;3409  R[0A]_K[.1]   ; Initialize SYMBOL_POINTER (to the
;3410       ; ...bits of the INFO_SYMBOLS vector)
```

```
U 1210, 0003,003C,0180,0988,0000,1211  ;3411  RC[1]_0     ; Clear CODE_VECTOR location
U 1211, 0810,0038,0180,0920,0000,1212  ;3412  D_RC[4]     ; Get INFO_SYMBOLS to be coded
U 1212, 0001,003C,0180,0990,0000,1213  ;3413  RC[2]_D     ; Initialize INFO_SYMBOLS
          ;3414  ALU_D[AND]R[0A],  ; See if BIT 0 of INFO_SYMBOLS is 0
U 1213, 000D,0034,0180,0A50,0010,1214  ;3415   CLK.UBCC  ; ...
U 1214, 0800,013C,0180,0A50,0000,10C2  ;3416  Z?,D_R[0A]   ; Get SYMBOL_POINTER
U 10C2, 001B,001C,1980,0988,0000,10C3  ;3417 =0 RC[1]_NOT.K[ZERO]  ; BIT 0 was NOT 0, set CODE_VECTOR to
          ;3418       ; ...all ones
U 10C3, 0021,003C,0180,0AD0,0000,10C4  ;3419  R[0A]_D.LEFT,SI/ZERO  ; Shift SYMBOL_POINTER left one bit
          ;3420       ; ...pointing to the next bit to
          ;3421       ; ...be encoded
U 10C4, 0000,003D,0180,0800,0000,1217  ;3422 =0 CALL[FIRST.ORDER]  ; Start encoding the first order
          ;3423       ; ...symbols into the final vector
U 10C5, 0000,003C,0180,0800,0000,10C6  ;3424  NOP
U 10C6, 0000,003D,0180,0800,0000,1223  ;3425 =0 CALL[SECOND.ORDER]  ; Start encoding the second order
          ;3426       ; ...symbols into the final vector
U 10C7, 0000,003E,0180,0800,0000,0001  ;3427  RETURN1    ; Return to caller with the encoded
          ;3428       ; ...vector in RC[1]
          ;3429
          ;3430
          ;3431
```

```
            ;3432  .PAGE
            ;3433  .TOC "    REED-MULLER VECTOR GENERATION ROUTINE"
            ;3434 ;*****************************************************************
            ;3435 ;*
            ;3436 ; THIS SUBROUTINE GENERATES THE VECTORS USED BY THE
            ;3437 ; REED-MULLER ALGORITHM TO GENERATE CODEWORDS.  THESE
            ;3438 ; VECTORS ARE:
            ;3439 ;
            ;3440 ;  V1: 10101010101010101010101010101010
            ;3441 ;  V2: 11001100110011001100110011001100
            ;3442 ;  V3: 11110000111100001111000011110000
            ;3443 ;  V4: 11111111000000001111111100000000
            ;3444 ;  V5: 11111111111111110000000000000000
            ;3445 ;
            ;3446 ; THE ROUTINE IS ARRANGED IN A TABLE, IN ORDER TO
            ;3447 ; BRANCH INTO IT WITH D3-0? BRANCH.
            ;3448 ;
            ;3449 ; **NOTE**: D<3:0> MUST HAVE THE VECTOR NUMBER WHEN
            ;3450 ;    CALLING THE ROUTINE.
            ;3451 ;
            ;3452 ; THE VECTOR IS RETURNED IN THE D-REGISTER.
            ;3453 ;
            ;3454 ; SIDE EFFECTS:
            ;3455 ;
            ;3456 ; The contents of the following registers will be lost:
            ;3457 ;
            ;3458 ; Q, SC
            ;3459 ;-
            ;3460 GENERATE.VECTOR:
U 1215, 0500,003C,0180,0800,0000,1216  ;3461  D_D.LEFT,SI/ZERO  ; Multiply VECTOR Pointer by 2
            ;3462       ; ... so as to index into the table
            ;3463       ; ...below
U 1216, 0000,193C,0180,0800,0000,1070  ;3464  D3-0?     ; Index into the Vector table
            ;3465 =0000 D_NOT.K[ZERO],    ; Generate V(0)
U 1070, 081B,001E,1980,0800,0000,0001  ;3466   RETURN1    ; ...
U 1071, 0000,003C,0180,0800,0000,1072  ;3467  NOP
U 1072, 001B,0000,1180,0800,0082,1073  ;3468 V1: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1073, 0F1B,0010,11C0,0800,0000,107C  ;3469  ALU_0+K[.4]+1,Q_ALU,D_0,J/VSHF  ; Q<3:0>=0101
U 1074, 001B,0000,1180,0800,0082,1075  ;3470 V2: ALU_0-K[.4],SC_ALU  ; SET SHIFT VALUE = -4
U 1075, 0F18,0038,0DC0,0800,0000,107C  ;3471  Q_K[.3],D_0,J/VSHF  ; Q<3:0>=0011
U 1076, 001B,0000,0180,0800,0082,1077  ;3472 V3: ALU_0-K[.8],SC_ALU  ; SET SHIFT VALUE = -8
U 1077, 0F18,0038,61C0,0800,0000,107C  ;3473  Q_K[.F],D_0,J/VSHF  ; Q<7:0>=00001111
U 1078, 001B,001C,6580,0800,0082,1079  ;3474 V4: ALU_NOT.K[.10],SC_ALU  ; K[SLOW],SHIFT VALUE = -16.
U 1079, 0F18,0038,49C0,0800,0080,D07C  ;3475  Q_K[.FF],D_0,SC_SC+1,J/VSHF ; Q<7:0>=11111111
U 107A, 0818,0038,C180,0800,0000,107F  ;3476 V5: D_K[.FFFF],J/V.EXIT
U 107B, 0000,003C,0180,0800,0000,107C  ;3477  NOP
U 107C, 0D00,003C,0180,0800,0000,107D  ;3478 VSHF: D_DAL.SC
U 107D, 0000,193C,0180,0800,0000,107E  ;3479  D0?     ; ANY V(I) IS DONE WHEN D<0>=1
U 107E, 0000,003C,0180,0800,0000,107C  ;3480 =1110 J/VSHF     ; LUCKY BR CONSTRAINT
            ;3481 V.EXIT:
            ;3482  D_NOT.D,    ; Adjust D as necessary
U 107F, 0801,002A,0180,0800,0000,0001  ;3483   RETURN1    ; Return to caller
            ;3484
```

```
;3485  .PAGE
;3486  .TOC "   FIRST ORDER REED-MULLER ENCODE"
;3487 ;*************************************************************
;3488 ;++
;3489 ;
;3490 ; Functional Description:
;3491 ; ----------------------
;3492 ;
;3493 ;      This subroutine is used by the ENCODE and DECODE
;3494 ; routines to encode the first order Reed-Muller vector
;3495 ; bits <05:01> of the received vector.
;3496 ;
;3497 ;
;3498 ; Calling Constraint: =0
;3499 ; ------------------
;3500 ;
;3501 ;
;3502 ; Inputs:
;3503 ; ------
;3504 ;
;3505 ; RC[2] must hold the vector to be encoded (INFO_SYMBOLS)
;3506 ;
;3507 ; R[0A] must hold the SYMBOL_POINTER as passed by
;3508 ; subroutines ENCODE or DECODE
;3509 ;
;3510 ; RC[1] must hold the CODE_VECTOR as passed by
;3511 ; subroutines ENCODE or DECODE
;3512 ;
;3513 ; Outputs:
;3514 ; -------
;3515 ;
;3516 ; RC[1] will contain upon exit the CODE_VECTOR for
;3517 ; bits <05:01> of INFO_SYMBOLS
;3518 ;
;3519 ; R[0A] will have an UPDATED SYMBOL_POINTER
;3520 ;
;3521 ;
;3522 ; Side Effects:
;3523 ; ------------
;3524 ;
;3525 ; The contents of the following registers will be lost
;3526 ;
;3527 ; D, Q, R[0], R[2]
;3528 ;
;3529 ;--
;3530 ;*************************************************************
;3531 FIRST.ORDER:
U 1217, 0018,0038,0980,0A80,0000,1218  ;3532  R[0]_K[.2]   ; Initialize the COUNTER
U 1218, 0018,0038,0580,0A90,0000,1219  ;3533  R[2]_K[.1]   ; Initialize variable J
        ;3534 FRST.ORDR.LOOP:
U 1219, 0810,0038,0180,0910,0000,121A  ;3535  D_RC[2]     ; Get INFO_SYMBOLS
        ;3536  ALU_D[AND]R[0A],  ; See if bit pointed by SYMBOL_POINTER
U 121A, 000D,0034,0180,0A50,0010,121B  ;3537   CLK.UBCC  ; ...in INFO_SYMBOLS is set
U 121B, 0000,013C,0180,0800,0000,10C8  ;3538  Z?
U 10C8, 0000,003C,0180,0800,0000,10CA  ;3539 =0 J/FRST.ORDR.SYM.NOT0  ; Bit was not 0
```

```
U 10C9, 0000,003C,0180,0800,0000,121D  ;3540  J/FRST.ORDR.SYM.EQ0  ; Bit was 0
          ;3541 =0
          ;3542 FRST.ORDR.SYM.NOT0:
          ;3543  D_R[2],    ; Point to VECTOR(J)
U 10CA, 0800,003D,0180,0A10,0000,1215  ;3544  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 10CB, 0010,0038,01C0,0908,0000,121C  ;3545  Q_RC[1]    ; Get CODE_VECTOR
          ;3546  ALU_D.XOR.Q,   ; Encode current bit pointed by
U 121C, 001D,0020,0180,0988,0000,121D  ;3547   RC[1]_ALU  ; ...SYMBOL_POINTER and save in RC[1]
          ;3548 FRST.ORDR.SYM.EQ0:
U 121D, 0800,003C,0180,0A10,0000,121E  ;3549  D_R[2]    ; Get J
U 121E, 0019,0014,0580,0A90,0000,121F  ;3550  R[2]_D+K[.1]   ; Increment J
U 121F, 0800,003C,0180,0A50,0000,1220  ;3551  D_R[0A]    ; Get SYMBOL_POINTER
U 1220, 0021,003C,D580,0AD0,0000,1221  ;3552  R[0A]_D.LEFT,SI/ZERO,K[.6] ; Point to the next bit to be encoded
          ;3553  ALU_R[0][XOR]K[.6],  ; Is the COUNTER equal to 6?
U 1221, 0018,0020,D580,0A00,0010,1222  ;3554   CLK.UBCC  ; ...(i.e., COUNTER .LE. 5)
          ;3555  Z?,    ;
U 1222, 0800,013C,0180,0A00,0000,10CC  ;3556   D_R[0]   ; Get the COUNTER just in case it has
          ;3557       ; ...to be incremented
          ;3558 =0 J/FRST.ORDR.LOOP,  ; Counter is NOT equal to 6 yet, thus
U 10CC, 0019,0014,0580,0A80,0000,1219  ;3559   R[0]_D+K[.1]  ; ...we did not encode all First
          ;3560       ; ...Order Symbols.  Repeat the loop
U 10CD, 0000,003E,0180,0800,0000,0001  ;3561  RETURN1    ; OK. COUNTER is equal to 6 so
          ;3562       ; ...we finished encoding ALL First
          ;3563       ; ...Order Symbols.  Return to caller.
          ;3564
          ;3565
```

```
;3566  .PAGE
;3567  .TOC "   SECOND ORDER REED-MULLER ENCODING ROUTINE"
;3568 ;*************************************************************
;3569 ;++
;3570 ;
;3571 ; Functional Description:
;3572 ; ----------------------
;3573 ;
;3574 ;      This subroutine is used by the ENCODE and DECODE
;3575 ; routines to encode the Second order Reed-Muller vector
;3576 ; of INFO_SYMBOLS vector.
;3577 ;
;3578 ;
;3579 ; Calling Constraint: =0
;3580 ; ------------------
;3581 ;
;3582 ;
;3583 ; Inputs:
;3584 ; ------
;3585 ;
;3586 ; RC[1] must hold the CODE_VECTOR as set up by FIRST.ORDER
;3587 ; routine or by the DECODE routine.
;3588 ;
;3589 ; RC[2] must hold the INFO_SYMBOLS vector (to be encoded)
;3590 ;
;3591 ; R[0A] must hold the SYMBOL_POINTER as set up by FIRST.ORDER
;3592 ; routine or by the DECODE routine.
;3593 ;
;3594 ;
;3595 ; Outputs:
;3596 ; -------
;3597 ;
;3598 ; RC[1] will hold upon exit the encoded vector
;3599 ;
;3600 ;
;3601 ; Side Effects:
;3602 ; ------------
;3603 ;
;3604 ; The contents of the following registers will be lost:
;3605 ;
;3606 ; D, Q, RC[1], R[0] through R[2], R[0A], SC
;3607 ;
;3608 ;--
;3609 ;*************************************************************
;3610 SECOND.ORDER:
U 1223. 0018,0038,05B0,0A80,0000,1224  ;3611  R[0]_K[.1]    ; Initialize COUNTER to 1
U 1224. 0F18,0038,1180,0A90,0000,1225  ;3612  R[2]_K[.4],D_0    ; Initialize variable J to 4
U 1225. 0019,0010,1180,0A88,0000,1226  ;3613  ALU_D+K[.4]+1,R[1]_ALU  ; Initialize variable I to 5
;3614 SCND.ORDR.LOOP:
U 1226. 0810,0038,0180,0910,0000,1227  ;3615  D_RC[2]     ; Get INFO_SYMBOLS
;3616   ALU_D[AND]R[0A],   ; See if bit pointed by SYMBOL_POINTER
U 1227. 000D,0034,0180,0A50,0010,1228  ;3617    CLK.UBCC   ; ...in INFO_SYMBOLS is set
U 1228. 0000,013C,0180,0800,0000,10CE  ;3618  Z?
U 10CE. 0000,003C,0180,0800,0000,10D0  ;3619 =0 J/SCND.ORDR.SYM.NOT0   ; Bit was not 0
U 10CF. 0000,003C,0180,0800,0000,122B  ;3620  J/SCND.ORDR.SYM.EQ0   ; Bit was 0
```

```
            ;3621 =0
            ;3622 SCND.ORDR.SYM.NOT0:
            ;3623  D_R[1],    ; Point to VECTOR(I)
U 10D0, 0800,003D,0180,0A08,0000,1215  ;3624  CALL[GENERATE.VECTOR]  ; ...and call routine to
U 10D1, 0001,003C,0180,0AF0,0000,10D2  ;3625  R[0E]_D    ; Save VECTOR(I)
            ;3626 =0 D_R[2],    ; Point to VECTOR(J)
U 10D2, 0800,003D,0180,0A10,0000,1215  ;3627  CALL[GENERATE.VECTOR]  ; ...and call routine to generate it
U 10D3, 0000,003C,01C0,0A70,0000,1229  ;3628  Q_R[0E]    ; Get saved VECTOR(I)
            ;3629  D_D.AND.Q,    ; "Combine" VECTOR(I) with VECTOR(J)
U 1229, 081D,0034,0180,0908,0000,122A  ;3630   LC_RC[1]  ; Get CODE_VECTOR
U 122A, 0011,0020,0180,0988,0000,122B  ;3631  RC[1]_D.XOR.LC   ; Encode bit pointed by SYMBOL_POINTER
            ;3632       ; ...in CODE_VECTOR
            ;3633 SCND.ORDR.SYM.EQ0:
            ;3634  ALU_R[2][A-B]K[.1],D_ALU, ; Decrement J
U 122B, 0818,0000,0580,0A10,0010,122C  ;3635   CLK.UBCC  ; Is it 0 yet?
            ;3636  Z?,
U 122C, 0000,013C,01C0,0A08,0000,10D4  ;3637   Q_R[1]   ; Get I
            ;3638 =0 J/SCND.ORDR.JNT0,  ; J is NOT 0.
U 10D4, 0001,003C,0180,0A90,0000,122E  ;3639   R[2]_D   ; Save J
U 10D5, 0019,2000,0580,0A90,0000,122D  ;3640  R[2]_Q-K[.1]   ; Decrement I and save
U 122D, 0019,2000,0580,0A88,0000,122E  ;3641  R[1]_Q-K[.1]   ; Reinitialize J to I - 1
            ;3642 SCND.ORDR.JNT0:
U 122E, 0800,003C,0180,0A50,0000,122F  ;3643  D_R[0A]    ; Get SYMBOL_POINTER
U 122F, 0021,003C,F5F8,0AD0,0000,1230  ;3644  R[0A]_D.LEFT,SI/ZERO,Q_Q,K[.A] ; Point to the next bit to be encoded
U 1230, 0819,2010,F580,0800,0000,1231  ;3645  ALU_Q+K[.A]+1,D_ALU  ; D = d11
            ;3646  ALU_D.XOR.R[0],    ; Is the COUNTER equal to d11?
U 1231, 000D,0020,0180,0A00,0010,1232  ;3647   CLK.UBCC  ; ...
            ;3648  Z?,     ; ...
U 1232, 0800,013C,0180,0A00,0000,10D6  ;3649   D_R[0]   ; Get the COUNTER in case it is
            ;3650       ; ...not equal to d11
            ;3651 =0 J/SCND.ORDR.LOOP,  ; COUNTER is NOT equal to d11 yet
U 10D6, 0019,0014,0580,0A80,0000,1226  ;3652   R[0]_D+K[.1]  ; Increment and save the COUNTER
U 10D7, 0000,003E,0180,0800,0000,0001  ;3653  RETURN1    ; Return to caller.
            ;3654       ; ...Finished encoding all second
            ;3655       ; ...order terms
            ;3656
```

```
                ;3657  .PAGE
                ;3658  .TOC "    REED-MULLER DECODE ROUTINE"
                ;3659 ;*************************************************************
                ;3660 ;**
                ;3661 ;
                ;3662 ; Functional Description:
                ;3663 ; ----------------------
                ;3664 ;
                ;3665 ;       This subroutine is used to decode the Reed-Muller
                ;3666 ; codes encoded by the ENCODE routine.
                ;3667 ;
                ;3668 ; Calling Constraint: =00
                ;3669 ; ------------------
                ;3670 ;
                ;3671 ;
                ;3672 ; Inputs:
                ;3673 ; ------
                ;3674 ;
                ;3675 ; RC[0] must hold the 32 bit vector to be decoded
                ;3676 ;
                ;3677 ;
                ;3678 ; Outputs:
                ;3679 ; -------
                ;3680 ;
                ;3681 ; RC[2] will hold the decoded vector
                ;3682 ;
                ;3683 ;
                ;3684 ; Side Effects:
                ;3685 ; ------------
                ;3686 ;
                ;3687 ;
                ;3688 ; See header of subroutine ENCODE
                ;3689 ;
                ;3690 ;
                ;3691 ;--
                ;3692 ;*************************************************************
                ;3693 DECODE:
U 1233.  0018.0038.6580.0A98.0000.1234  ;3694  R[3]_K[.10]    ; Initialize variable K to d16
U 1234.  0018.0038.6580.0AA0.0000.1235  ;3695  R[4]_K[.10]    ; Initialize variable L to d16
U 1235.  0003.003C.0180.0990.0000.1236  ;3696  RC[2]_0     ; Initialize variable INFO_SYMBOLS to 0
U 1236.  0810.0038.0180.0900.0000.1237  ;3697  D_RC[0]     ; Get the RCVD_VECTOR
U 1237.  0001.003C.0180.0AB0.0000.1238  ;3698  R[6]_D     ; Save RCVD_VECTOR
                ;3699 DECODE.SCND.ORDR.LOOP:
U 1238.  0840.003C.0180.0A20.0000.1239  ;3700  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; Divide L by 2
U 1239.  0001.003C.0180.0AA0.0000.123A  ;3701  R[4]_D     ; L = L/2
U 123A.  0003.003C.0180.0AC0.0000.123B  ;3702  R[0]_0     ; Initialize variable 1ST to 0
U 123B.  0003.003C.0180.0AC8.0000.123C  ;3703  R[9]_0     ; Initialize variable SAVE_1ST to 0
U 123C.  0003.003C.0180.0998.0000.123D  ;3704  RC[3]_0     ; Initialize varaible COVERED_BITS to 0
U 123D.  0018.0038.0580.0A80.0000.123E  ;3705  R[0]_K[.1]    ; Initialize COUNTER to 1
U 123E.  0003.003C.0180.0AB8.00C0.123F  ;3706  R[7]_0     ; Initialize variable ONE_COUNTER to 0
                ;3707 DECODE.SEARCH.LOOP:
U 123F.  0003.003C.0180.0AA8.0000.1240  ;3708  R[5]_0     ; Initialize variable SUM to 0
U 1240.  0003.003C.0180.0AD8.0000.1241  ;3709  R[0B]_0     ; Initialize variable CURRENT_BIT_MASK
                ;3710          ; ...to 0
U 1241.  0000.003C.0180.0A40.0082.1242  ;3711  SC_R[8]     ; Get 1ST
```

```
U 1242. 0003,001C,0180,0AD8,0000,1243  ;3712  R[0B]_NOT.MASK   ; Set bit 1ST in CURRENT_BIT_MASK
        ;3713  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST set in RCVD_VECTOR?
U 1243. 0000,0024,0180,0A30,0010,1244  ;3714   CLK.UBCC  ; ...
U 1244. 0800,013C,0180,0A28,0000,10D8  ;3715  Z?,D_R[5]   ; Get SUM
U 10D8. 0019,0014,0580,0AA8,0000,10D9  ;3716 =0 R[5]_D+K[.1]   ; Bit 1ST in RCVD_VECTOR was NOT 0,
        ;3717       ; ...increment the SUM
U 10D9. 0800,003C,0180,0A40,0000,1245  ;3718  D_R[8]    ; Get 1ST
U 1245. 000D,0014,01C0,0A20,0082,1246  ;3719  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 1246. 0800,001C,0180,0A58,0000,1247  ;3720  D_R[0B].ORNOT.MASK  ; Set bit 1ST + L in CURRENT_BIT_MASK
U 1247. 0001,003C,0180,0AD8,0000,1248  ;3721  R[0B]_D    ; Save CURRENT_BIT_MASK
        ;3722  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + L set in RCVD_VECTOR?
U 1248. 0000,0024,0180,0A30,0010,1249  ;3723   CLK.UBCC  ; ...
U 1249. 0800,013C,0180,0A28,0000,10DA  ;3724  Z?,D_R[5]   ; Get SUM
U 10DA. 0019,0014,0580,0AA8,0000,10DB  ;3725 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 10DB. 0800,003C,0180,0A58,0000,124A  ;3726  D_R[0B]    ; Get CURRENT_BIT_MASK
U 124A. 000D,2014,0180,0A18,0082,124B  ;3727  ALU_Q[A+B]R[3],SC_ALU  ; SC = 1ST + K + L
U 124B. 0001,001C,0180,0AD8,0000,124C  ;3728  R[0B]_D.ORNOT.MASK  ; Set bit 1ST + K + L in
        ;3729       ; ...CURRENT_BIT_MASK
        ;3730  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + K + L set
U 124C. 0000,0024,0180,0A30,0010,124D  ;3731   CLK.UBCC  ; ...in RCVD_VECTOR?
U 124D. 0800,013C,0180,0A28,0000,10DC  ;3732  Z?,D_R[5]   ; Get SUM
U 10DC. 0019,0014,0580,0AA8,0000,10DD  ;3733 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 10DD. 0800,003C,0180,0A40,0000,124E  ;3734  D_R[8]    ; Get 1ST
U 124E. 000D,0014,0180,0A18,0082,124F  ;3735  ALU_D[A+B]R[3],SC_ALU  ; SC = 1ST + K
U 124F. 0800,001C,0180,0A58,0000,1250  ;3736  D_R[0B].ORNOT.MASK  ; Set bit 1ST + K in CURRENT_BIT_MASK
U 1250. 0001,003C,0180,0AD8,0000,1251  ;3737  R[0B]_D    ; Save CURRENT_BIT_MASK
        ;3738  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + K set in RCVD_VECTOR
U 1251. 0000,0024,0180,0A30,0010,1252  ;3739   CLK.UBCC  ; ...
U 1252. 0800,013C,0180,0A28,0000,10DE  ;3740  Z?,D_R[5]   ; Get SUM
U 10DE. 0019,0014,0580,0AA8,0000,10DF  ;3741 =0 R[5]_D+K[.1]   ; Bit was NOT 0. Increment SUM
U 10DF. 0800,003C,0180,0A58,0000,1253  ;3742  D_R[0B]    ; Get CURRENT_BIT_MASK
        ;3743  ALU_D[AND]RC[3],   ; Are bits set in CURRENT_BIT_MASK
U 1253. 0011,0034,0180,0918,0010,1254  ;3744   CLK.UBCC  ; ...also set in COVERED_BITS
U 1254. 0000,013C,0180,0800,0000,10E0  ;3745  Z?
        ;3746 =0 J/DECODE.BITS.COVERED,   ; Some of the bits in CURRENT_BIT_MASK
U 10E0. 0820,003C,0180,0A20,0000,1255  ;3747  D_R[4].LEFT,SI/ZERO  ; ...have been previously covered.
        ;3748       ; ...D = 2 * L
        ;3749  J/DECODE.BITS.NOT.COVERED, ; Bits in CURRENT_BIT_MASK were not
U 10E1. 0818,0014,0580,0A40,0000,1258  ;3750  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
        ;3751 DECODE.BITS.COVERED:
U 1255. 080D,0014,0180,0A48,0000,1256  ;3752  ALU_D[A+B]R[9],D_ALU  ; D = SAVE_1ST + (2 * L)
U 1256. 0001,003C,0180,0AC0,0000,1257  ;3753  R[8]_D    ; 1ST = SAVE_1ST + (2 * L)
        ;3754  R[9]_D,    ; SAVE_1ST = SAVE_1ST + (2 * L)
U 1257. 0001,003C,0180,0AC8,0000,123F  ;3755   J/DECODE.SEARCH.LOOP ; Repeat the search loop for
        ;3756       ; ...unCOVERED_BITS
        ;3757 DECODE.BITS.NOT.COVERED:
U 1258. 0001,003C,0180,0AC0,0000,1259  ;3758  R[8]_D    ; 1ST = 1ST + 1
U 1259. 0800,003C,0180,0A58,0000,125A  ;3759  D_R[0B]    ; Get CURRENT_BIT_MASK
U 125A. 0811,0030,0180,0918,0000,125B  ;3760  D_D.OR.RC[3]   ; COVERED_BITS = COVERED_BITS .OR.
        ;3761       ; ...CURRENT_BIT_MASK
U 125B. 0001,003C,0180,0998,0000,125C  ;3762  RC[3]_D    ; Save COVERED_BITS mask
U 125C. 0000,003C,01C0,0A38,0000,125D  ;3763  Q_R[7]    ; Get ONE_COUNTER
U 125D. 0800,003C,0180,0A28,0000,125E  ;3764  D_R[5]    ; Get SUM and see if it is EVEN or ODD
U 125E. 0000,193C,0180,0800,0000,108E  ;3765  D1-0?    ; ...
U 108E. 0000,003C,0180,0800,0000,125F  ;3766 =1110 J/DECODE.SUM.EVEN  ; SUM was even
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                                    SEQ 1218
ESKAR4A.MCR                    MICRO2  1P(00)    26-JUL-85  17:26:47                                   Page  10

U 108F. 0019,2014,0580,0AB8,0000,125F  ;3767  R[7]_Q+K[.1]   ; SUM is ODD.  Increment ONE_COUNTER
          ;3768 DECODE.SUM.EVEN:
          ;3769  ALU_R[0].XOR.K[.8],  ; Is COUNTER equal to 8?
U 125F. 0018,0020,0180,0A00,0010,1260  ;3770   CLK.UBCC  ; ...
U 1260. 0800,013C,0180,0A00,0000,10E2  ;3771  Z?,D_R[0]   ; Get COUNTER in case it has to be
          ;3772       ; ...incremented
          ;3773 =0 J/DECODE.SEARCH.LOOP,  ; COUNTER is NOT equal to 8 yet
U 10E2. 0019,0014,0580,0A80,0000,123F  ;3774   R[0]_D+K[.1]  ; Increment COUNTER
          ;3775  ALU_R[7][A-B]K[.4],  ; Is ONE_COUNTER GREATER than or
U 10E3. 0018,0000,1180,0A38,0010,1261  ;3776   CLK.UBCC  ; ...EQUAL to  4?
          ;3777  ALU.N?,    ; Check Condition Codes.
          ;3778   D_RC[2],  ; Get INFO_SYMBOLS
U 1261. 0810,1B38,4580,0910,0000,1047  ;3779   K[.8000]  ; Bring up slow constant
U 1047. 0819,0030,4580,0800,0000,104F  ;3780 =0111 D_D.OR.K[.8000]   ; ONE_COUNTER GREATER than or
          ;3781       ; ...EQUAL to 4.  Set bit 0 of
          ;3782       ; ...INFO_SYMBOLS
          ;3783  RC[2]_ALU.RIGHT,SI/ZERO, ; Shift INFO_SYMBOLS right and Save
U 104F. 0041,003C,0180,0990,0000,1262  ;3784   ALU_D   ; ...
          ;3785  ALU_R[4][XOR]K[.1],  ; Is L EQUAL to 1?
U 1262. 0018,0020,0580,0A20,0010,1263  ;3786   CLK.UBCC  ; ...
          ;3787  Z?,
U 1263. 0840,013C,0180,0A18,0000,10E4  ;3788  D_ALU.RIGHT,ALU_R[3],SI/ZERO ; D = K/2. In case L = 1
U 10E4. 0000,003C,0180,0800,0000,1238  ;3789 =0 J/DECODE.SCND.ORDR.LOOP  ; L was not EQUAL to 1. Not all SECOND
          ;3790       ; ...ORDER symbols have been decoded.
          ;3791       ; ...Repeat loop.
          ;3792  ALU_D.XOR.K[.1],  ; Is K = 1?
U 10E5. 0019,0020,0580,0800,0010,1264  ;3793   CLK.UBCC  ; ...
U 1264. 0001,013C,0180,0A98,0000,10E6  ;3794  Z?,R[3]_D   ; Save K just in case.
          ;3795 =0 J/DECODE.SCND.ORDR.LOOP, ; OK. K was updated and it is NOT equal
U 10E6. 0001,003C,0180,0AA0,0000,1238  ;3796   R[4]_D   ; ...to 1.  Update L.  L = K/2
U 10E7. 0810,0038,0180,0910,0000,1265  ;3797  D_RC[2]    ; Get INFO_SYMBOLS
U 1265. 0200,003C,0180,0800,0000,1266  ;3798  D_D.RIGHT2,SI/ZERO  ; Shift INFO_SYMBOLS right 4 bits
U 1266. 0200,003C,0180,0800,0000,1267  ;3799  D_D.RIGHT2,SI/ZERO  ; ...
U 1267. 0001,003C,0180,0990,0000,1268  ;3800  RC[2]_D    ; Save INFO_SYMBOLS
U 1268. 0018,0038,0980,0AD0,0000,1269  ;3801  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 1269. 0003,003C,0180,0988,0000,10E8  ;3802  RC[1]_0    ; Clear CODE_VECTOR
U 10E8. 0000,003D,0180,0800,0000,1223  ;3803 =0 CALL[SECOND.ORDER]  ; Re-encode the second order symbols
          ;3804       ; ...so as to extract them from
          ;3805       ; ...RCVD_VECTOR
U 10E9. 0800,003C,0180,0A30,0000,126A  ;3806  D_R[6]    ; Get RCVD_VECTOR
U 126A. 0811,0020,0180,0908,0000,126B  ;3807  D_D.XOR.RC[1]   ; "Subtract" the encoded second order
          ;3808       ; ...symbols from RCVD_VECTOR
U 126B. 0001,003C,0180,0AB0,0000,126C  ;3809  R[6]_D    ; Save the result
U 126C. 0018,0038,7580,0AA0,0000,126D  ;3810  R[4]_K[.20]   ; Initialize variable L to d32
          ;3811 DECODE.FRST.ORDER.LOOP:
U 126D. 0840,003C,0180,0A20,0000,126E  ;3812  D_ALU.RIGHT,ALU_R[4],SI/ZERO ; D = L/2
U 126E. 0001,003C,0180,0AA0,0000,126F  ;3813  R[4]_D    ; Save L
U 126F. 0003,003C,0180,0AC0,0000,1270  ;3814  R[8]_0    ; Initialize variable 1ST
U 1270. 0003,003C,0180,0AC8,0000,1271  ;3815  R[9]_0    ; Initialize variable SAVE_1ST
U 1271. 0003,003C,0180,0998,0000,1272  ;3816  RC[3]_0    ; Initialize variable COVERED_BITS to 0
U 1272. 0018,0038,0580,0A80,0000,1273  ;3817  R[0]_K[.1]   ; Initialize COUNTER to 1
U 1273. 0003,003C,0180,0AB8,0000,1274  ;3818  R[7]_0    ; Clear ONE_COUNTER
          ;3819 DECODE.FRST.ORDR.SEARCH:
U 1274. 0003,003C,0180,0AA8,0000,1275  ;3820  R[5]_0    ; Initialize variable SUM to 0
U 1275. 0003,003C,0180,0AD8,0000,1276  ;3821  R[0B]_0    ; Initialize variable CURRENT_BIT_MASK
```

```
            ;3822       ; ...to 0
U 1276, 0000,003C,0180,0A40,0082,1277  ;3823  SC_R[8]     ; Get 1ST
U 1277, 0003,001C,0180,0AD8,0000,1278  ;3824  R[0B]_NOT.MASK    ; Set bit 1ST in CURRENT_BIT_MASK
            ;3825  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST set in RCVD_VECTOR?
U 1278, 0000,0024,0180,0A30,0010,1279  ;3826    CLK.UBCC  ; ...
U 1279, 0800,013C,0180,0A28,0000,10EA  ;3827  Z?,D_R[5]    ; Get SUM
U 10EA, 0019,0014,0580,0AA8,0000,10EB  ;3828 =0 R[5]_D+K[.1]    ; Bit 1ST in RCVD_VECTOR was NOT 0,
            ;3829       ; ...increment the SUM
U 10EB, 0800,003C,0180,0A40,0000,127A  ;3830  D_R[8]     ; Get 1ST
U 127A, 000D,0014,01C0,0A20,0082,127B  ;3831  ALU_D[A+B]R[4],SC_ALU,Q_ALU ; Q = 1ST + L
U 127B, 0800,001C,0180,0A58,0000,127C  ;3832  D_R[0B].ORNOT.MASK   ; Set bit 1ST + L in CURRENT_BIT_MASK
U 127C, 0001,003C,0180,0AD8,0000,127D  ;3833  R[0B]_D     ; Save CURRENT_BIT_MASK
            ;3834  ALU_R[6].ANDNOT.MASK,   ; Is bit 1ST + L set in RCVD_VECTOR?
U 127D, 0000,0024,0180,0A30,0010,127E  ;3835    CLK.UBCC  ; ...
U 127E, 0800,013C,0180,0A28,0000,10EC  ;3836  Z?,D_R[5]    ; Get SUM
U 10EC, 0019,0014,0580,0AA8,0000,10ED  ;3837 =0 R[5]_D+K[.1]    ; Bit was NOT 0. Increment SUM
U 10ED, 0800,003C,0180,0A58,0000,127F  ;3838  D_R[0B]     ; Get CURRENT_BIT_MASK
            ;3839  ALU_D[AND]RC[3],    ; Are bits set in CURRENT_BIT_MASK
U 127F, 0011,0034,0180,0918,0010,1280  ;3840    CLK.UBCC  ; ...also set in COVERED_BITS
U 1280, 0000,013C,0180,0800,0000,10EE  ;3841  Z?
            ;3842 =0 J/DECODE.BITS.COVRD,   ; Some of the bits in CURRENT_BIT_MASK
U 10EE, 0820,003C,0180,0A20,0000,1281  ;3843  D_R[4].LEFT,SI/ZERO  ; ...have been previously covered.
            ;3844       ; ...D = 2 * L
            ;3845  J/DECODE.BITS.NOT.COVRD, ; Bits in CURRENT_BIT_MASK were not
U 10EF, 0818,0014,0580,0A40,0000,1284  ;3846  ALU_R[8][A+B]K[.1],D_ALU ; ...covered, D = 1ST + 1
            ;3847 DECODE.BITS.COVRD:
U 1281, 080D,0014,0180,0A48,0000,1282  ;3848  ALU_D[A+B]R[9],D_ALU  ; D = SAVE_1ST + (2 * L)
U 1282, 0001,003C,0180,0AC0,0000,1283  ;3849  R[8]_D     ; 1ST = SAVE_1ST + (2 * L)
            ;3850  R[9]_D,     ; SAVE_1ST = SAVE_1ST + (2 * L)
U 1283, 0001,003C,0180,0AC8,0000,1274  ;3851    J/DECODE.FRST.ORDR.SEARCH ; Repeat the search loop for
            ;3852       ; ...unCOVERED_BITS
            ;3853 DECODE.BITS.NOT.COVRD:
U 1284, 0001,003C,0180,0AC0,0000,1285  ;3854  R[8]_D     ; 1ST = 1ST + 1
U 1285, 0800,003C,0180,0A58,0000,1286  ;3855  D_R[0B]     ; Get CURRENT_BIT_MASK
U 1286, 0811,0030,0180,0918,0000,1287  ;3856  D_D.OR.RC[3]    ; COVERED_BITS = COVERED_BITS .OR.
            ;3857       ; ...CURRENT_BIT_MASK
U 1287, 0001,003C,0180,0998,0000,1288  ;3858  RC[3]_D     ; Save COVERED_BITS mask
U 1288, 0000,003C,01C0,0A38,0000,1289  ;3859  Q_R[7]     ; Get ONE_COUNTER
U 1289, 0800,003C,0180,0A28,0000,128A  ;3860  D_R[5]     ; Get SUM and see if it is EVEN or ODD
U 128A, 0000,193C,0180,0800,0000,109E  ;3861  D1-0?     ; ...
U 109E, 0000,003C,0180,0800,0000,128B  ;3862 =1110 J/DECODE.SUM.EVEN2  ; SUM was even
U 109F, 0019,2014,0580,0AB8,0000,128B  ;3863  R[7]_Q+K[.1]    ; SUM is ODD.  Increment ONE_COUNTER
            ;3864 DECODE.SUM.EVEN2:
            ;3865  ALU_R[0].XOR.K[.10],    ; Is COUNTER equal to d16?
U 128B, 0018,0020,6580,0A00,0010,128C  ;3866    CLK.UBCC  ; ...
U 128C, 0800,013C,0180,0A00,0000,10F0  ;3867  Z?,D_R[0]    ; Get COUNTER in case it has to be
            ;3868       ; ...incremented
            ;3869 =0 J/DECODE.FRST.ORDR.SEARCH,  ; COUNTER is NOT equal to 16 yet
U 10F0, 0019,0014,0580,0A80,0000,1274  ;3870    R[0]_D+K[.1]    ; Increment COUNTER
            ;3871  ALU_R[7][A-B]K[.8],    ; Is ONE_COUNTER GREATER than or
U 10F1, 0018,0000,0180,0A38,0010,128D  ;3872    CLK.UBCC  ; ...EQUAL to  8?
            ;3873  ALU.N?,     ; Check Condition Codes.
U 128D, 0810,1B38,0180,0910,0000,1057  ;3874    D_RC[2]    ; Get INFO_SYMBOLS
U 1057, 0819,0030,0580,0800,0000,105F  ;3875 =0111 D_D.OR.K[.1]    ; ONE_COUNTER GREATER than or
            ;3876       ; ...EQUAL to 8.  Set bit 0 of
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                                    SEQ 1220
ESKAR4A.MCR                    MICRO2  1P(00)    26-JUL-85  17:26:47                                   Page  10

            ;3877       ; ...INFO_SYMBOLS
U 105F, 0021,003C,0180,0990,0000,128E  ;3878  RC[2]_D.LEFT,SI/ZERO  ; Shift INFO_SYMBOLS left and Save
            ;3879  ALU_R[4].XOR.K[.1],  ; Is L = 1?
U 128E, 0018,0020,0580,0A20,0010,128F  ;3880   CLK.UBCC
U 128F, 0000,013C,0180,0800,0000,10F2  ;3881  Z?
U 10F2, 0000,003C,0180,0800,0000,126D  ;3882 =0 J/DECODE.FRST.ORDER.LOOP ; L is NOT equal to 1 yet. Repeat loop.
U 10F3, 0003,003C,0180,0988,0000,1290  ;3883  RC[1]_0     ; Initialize CODE_VECTOR to 0
U 1290, 0018,0038,0980,0AD0,0000,10F4  ;3884  R[0A]_K[.2]   ; Set bit 1 in SYMBOL_POINTER
U 10F4, 0000,003D,0180,0800,0000,1217  ;3885 =0 CALL[FIRST.ORDER]  ; Generate the first order Reed-Muller
            ;3886       ; ...code for the symbols just
            ;3887       ; ...decoded
U 10F5, 0800,003C,0180,0A30,0000,1291  ;3888  D_R[6]     ; Get RCVD_VECTOR
U 1291, 0010,0038,01C0,090B,0000,1292  ;3889  Q_RC[1]     ; Get CODE_VECTOR
U 1292, 081D,2020,0180,0AB0,0000,1293  ;3890  Q_Q.XOR.D,R[6]_ALU  ; RCVD_VECTOR = RCVD_VECTOR .XOR.
            ;3891       ; ... CODE_VECTOR
U 1293, 0003,003C,0180,0ABB,0000,1294  ;3892  R[7]_0     ; Initialize ONE_COUNTER to 0
U 1294, 0018,0038,0580,0AD0,0000,1295  ;3893  R[0A]_K[.1]   ; Set Bit 0 of SYMBOL_POINTER
            ;3894 DECODE.ZERO.ORDR.LOOP:
U 1295, 0800,003C,0180,0A50,0000,1296  ;3895  D_R[0A]     ; Get SYMBOL_POINTER
            ;3896  ALU_D[AND]R[6],    ; Is SYMBOL_POINTER .AND. RCVD_VECTOR
U 1296, 000D,0034,0180,0A30,0010,1297  ;3897   CLK.UBCC  ; ...equal to 0?
            ;3898  Z?,
U 1297, 0800,013C,0180,0A38,0000,10F6  ;3899   D_R[7]    ; Get ONE_COUNTER
            ;3900 =0 J/DECODE.ZERO.ORDR.BITNZ, ; Bit was NOT 0 in RCVD_VECTOR
U 10F6, 0019,0014,0580,0AB8,0000,1298  ;3901   R[7]_D+K[.1]  ; Increment ONE_COUNTER
            ;3902  J/DECODE.ZERO.ORDR.BITZ, ; Bit was equal to 0
U 10F7, 0800,003C,0180,0A50,0000,129B  ;3903   D_R[0A]    ; Get SYMBOL_POINTER
            ;3904 DECODE.ZERO.ORDR.BITNZ:
U 1298, 0800,003C,0180,0A30,0000,1299  ;3905  D_R[6]     ; Get RCVD_VECTOR
            ;3906  ALU_D[ANDNOT]R[0A],D_ALU, ; RCVD_VECTOR = RCVD_VECTOR .ANDNOT.
U 1299, 080D,0024,0180,0A50,0010,129A  ;3907   CLK.UBCC  ; ...SYMBOL_POINTER (i.e., clear bit
            ;3908       ; ...tested and see if the overall
            ;3909       ; ...result is 0)
U 129A, 0000,013C,0180,0800,0000,10F8  ;3910  Z?
            ;3911 =0 J/DECODE.ZERO.ORDR.BITZ, ; RCVD_VECTOR is NOT zero so
U 10F8, 0800,003C,0180,0A50,0000,129B  ;3912   D_R[0A]    ; ...continue decoding all zero
            ;3913       ; ...order bits
            ;3914  J/DECODE.ZERO.ORDR.DONE, ; RCVD_VECTOR is 0 so we finished
U 10F9, 0800,003C,0180,0A38,0000,129D  ;3915   D_R[7]    ; ...decoding all zero order bits.
            ;3916 DECODE.ZERO.ORDR.BITZ:
            ;3917  R[0A]_D.LEFT,SI/ZERO,   ; Arithmetic Shift Left SYMBOL_POINTER.
U 129B, 0021,003C,0180,0AD0,0010,129C  ;3918   CLK.UBCC  ; ...and see if it is 0
U 129C, 0000,013C,0180,0800,0000,10FA  ;3919  Z?
U 10FA, 0000,003C,0180,0800,0000,1295  ;3920 =0 J/DECODE.ZERO.ORDR.LOOP  ; ...Continue decoding
U 10FB, 0800,003C,0180,0A38,0000,129D  ;3921  D_R[7]     ; Get ONE_COUNTER
            ;3922 DECODE.ZERO.ORDR.DONE:
            ;3923  ALU_D-K[.10],    ; Is ONE_COUNTER GREATER than or EQUAL
U 129D, 0019,0000,6580,0800,0010,129E  ;3924   CLK.UBCC  ; ...to d16.
            ;3925  ALU.N?,
U 129E, 0810,1B38,0180,0910,0000,10A7  ;3926   D_RC[2]    ; Get INFO_SYMBOLS
U 10A7, 0019,0030,0580,0990,0000,10AF  ;3927 =0111 RC[2]_D.OR.K[.1]  ; Set Bit 0 of INFO_SYMBOLS
U 10AF, 0000,003C,0180,0800,0000,129F  ;3928  NOP
            ;3929 ;
            ;3930 ;-
            ;3931 ;
```

```
            ;3932 ; This is where the actual decoding of the received Reed-Muller
            ;3933 ; code (RCVD_VECTOR) ends.  The decoded result is left in
            ;3934 ; RC[2]  (INFO_SYMBOLS) which will be now compared against
            ;3935 ; the expected value which is in RC[4].
            ;3936 ;
            ;3937 ;**
            ;3938 ;
U 129F, 0810,0038,0180,0910,0000,12A0  ;3939  D_RC[2]    ; Get INFO_SYMBOLS
            ;3940  ALU_D[XOR]RC[4],  ; See if INFO_SYMBOLS is equal to
U 12A0, 0011,0020,0180,0920,0010,12A1  ;3941   CLK.UBCC  ; ...the expected data
U 12A1, 0000,013C,0180,0800,0000,10FC  ;3942  Z?
            ;3943 =0 J/DECODE.ERROR,   ; Data Returned is not equal to
U 10FC, 0000,003C,0180,0800,1480,12A2  ;3944  SC_STATE   ; ...the expected.  Get number of
            ;3945       ; ...errors
U 10FD, 0000,003E,0180,0800,0000,0003  ;3946  RETURN3    ; OK. No error in data so do a
            ;3947       ; ...RETURN3
            ;3948 DECODE.ERROR:
U 12A2, 0000,003C,E980,0800,0084,92A3  ;3949  SC_SC+K[.24]   ; Get number of errors and transform it
            ;3950       ; ...in an ID Register address
            ;3951       ; ...starting with ID register 25
U 12A3, 0000,003C,0580,0800,0084,92A4  ;3952  SC_SC+K[.1]   ; SC Reg holds a pointer to ID regs 25
            ;3953       ; ...through 2X depending on how many
            ;3954       ; ...errors occurred.
U 12A4, 0810,0038,0180,0910,0000,12A5  ;3955  D_RC[2]    ; Get Received DECODED data
U 12A5, 0000,003C,0180,3400,0000,12A6  ;3956  ID(SC)_D   ; Save it in ID Reg pointed by SC
            ;3957  D_RC[4],   ; Get Expected Data
U 12A6, 0810,0038,0580,0920,0084,92A7  ;3958   SC_SC+K[.1]  ; Increment pointer to ID Regs
U 12A7, 0000,003C,0180,3400,0000,12A8  ;3959  ID(SC)_D   ; Save Expected data in ID reg
U 12A8, 0000,003C,0980,0800,1404,92A9  ;3960  STATE_STATE+K[.2]  ; Increment the error counter.
U 12A9, 0000,003C,0180,0800,1480,12AB  ;3961  SC_STATE   ; Get the number of errors
U 12AB, 0818,0038,1D80,0800,0000,12AC  ;3962  D_SC    ; ...in D reg
            ;3963  ALU_D.XOR.K[.C],  ; Is the error counter equal to
U 12AC, 0019,0020,8580,0800,0010,12AD  ;3964   CLK.UBCC  ; ...x0C? (i.e., maximum number of
            ;3965       ; ...errors has been reached)
U 12AD, 0000,013C,0180,0800,0000,10FE  ;3966  Z?
U 10FE, 0000,003E,0180,0800,0000,0002  ;3967 =0 RETURN2    ; Maximum number of errors has NOT been
            ;3968       ; ...reached yet
U 10FF, 0000,003E,0180,0800,0000,0001  ;3969  RETURN1    ; Maximum number of errors!!
            ;3970
            ;3971
            ;3972
```

```
;3973  .PAGE
;3974  .TOC "   FIND CONTROLLERS"
;3975 ;*****************************************************************
;3976 ;**
;3977 ;
;3978 ; Functional Description:
;3979 ; ----------------------
;3980 ;
;3981 ;
;3982 ;       This subroutine is called by tests that  need  to  know  how
;3983 ; many  controllers  are available on the MS780-E/H currently under
;3984 ; test.  It makes  use  of  subroutine  SELECT.CNTRLR  to  put  the
;3985 ; controllers in different Interleave Modes.
;3986 ; The subroutine returns control to the calling routine  by  either
;3987 ; one  of:   Return1, Return2, Return3 or Return4.  It is up to the
;3988 ; test to decode this information as necessary.
;3989 ; Since some Tests require this feature, the subroutine also  finds
;3990 ; the  memory  size  available  on each controller and leaves it in
;3991 ; registers R[0] and R[1].
;3992 ;
;3993 ;
;3994 ; Calling Constraint: =000
;3995 ; ------------------
;3996 ;
;3997 ;
;3998 ; Inputs:
;3999 ; ------
;4000 ;
;4001 ; RC[0E] MUST HOLD THE I/O BASES ADDRESS OF THE MS780-E/H
;4002 ; UNDER TEST
;4003 ;
;4004 ; Outputs:
;4005 ; -------
;4006 ;
;4007 ; - Return1, Return2, Return3 or Return4 depending on
;4008 ;   how many Controllers were found
;4009 ;
;4010 ; - R[0] Size of Memory on the Lower Controller
;4011 ;   R[1] Size of Memory on the Upper Controller
;4012 ;
;4013 ; Side Effects:
;4014 ; ------------
;4015 ;
;4016 ; The contents of the following registers will be lost:
;4017 ;
;4018 ; D, Q, SC, R[0], R[1]
;4019 ;
;4020 ;--
;4021 ;*****************************************************************
;4022 =00
;4023 FIND.CNTRLRS:
;4024  D_K[ZERO],    ; Attempt to select the Lower
U 1034, 0818.0039,1980,0800,0000,116E  ;4025   CALL[SELECT.CNTRLR] ; ...Controller
U 1035, 0000.003C,0180,0800,0000,1048  ;4026  J/LOW.MIS.X    ; Lower Controller Misconfigured
U 1036, 0000.003D,0180,0800,0000,11E2  ;4027  CALL[GET.MEMORY.SIZE]  ; Lower Controller has been  configured
```

```
            ;4028       ; ...find its memory size
U 1037, 0001,003C,0180,0A80,0000,1040  ;4029  R[0]_D     ; Save Memory size in R[0]
            ;4030 =00 D_K[.2],    ; Attempt to select the Upper
U 1040, 0818,0039,0980,0800,0000,116E  ;4031    CALL[SELECT.CNTRLR] ; ...Controller
U 1041, 0000,003E,0180,0800,0000,0002  ;4032  RETURN2     ; Upper Controller Misconfigured.
            ;4033       ; ...Let caller know that ONLY the
            ;4034       ; ...Lower Controller was found
U 1042, 0000,003D,0180,0800,0000,11E2  ;4035  CALL[GET.MEMORY.SIZE]  ; Upper Controller was Configured so
            ;4036       ; ...find its memory size
            ;4037  R[1]_D,     ; Save Memory Size in R[1]
U 1043, 0001,003E,0180,0A88,0000,0004  ;4038    RETURN4    ; Let Caller know that BOTH Controllers
            ;4039       ; ...were found
            ;4040 =00
            ;4041 LOW.MIS.X:
            ;4042  D_K[.2],    ; Lower Controller was not found,
U 1048, 0818,0039,0980,0800,0000,116E  ;4043    CALL[SELECT.CNTRLR] ; ...so attempt to Configure the
            ;4044       ; ...Upper Controller
U 1049, 0000,003E,0180,0800,0000,0001  ;4045  RETURN1     ; Tell Caller that NONE of the
            ;4046       ; ...Controllers were found
U 104A, 0000,003D,0180,0800,0000,11E2  ;4047  CALL[GET.MEMORY.SIZE]  ; Upper Controller was found,
            ;4048       ; ...go get its memory size
            ;4049  R[1]_D,     ; Save the memory size in R[1]
U 104B, 0001,003E,0180,0A88,0000,0003  ;4050    RETURN3    ; Let Caller know that ONLY the
            ;4051       ; ...Upper Controller was found
            ;4052
            ;4053
```

```
            ;4054  .PAGE
            ;4055  .TOC "   DELAY"
            ;4056 ;*****************************************************************
            ;4057 ;**
            ;4058 ;
            ;4059 ; Functional Description:
            ;4060 ; ----------------------
            ;4061 ;
            ;4062 ; This subroutine is used to reset the Time-Out counter in
            ;4063 ; Mic-Mon.  This may be desired when a test is running a loop
            ;4064 ; and does not have to do any calls to Mic-Mon.
            ;4065 ;
            ;4066 ;
            ;4067 ; Calling Constraint: =0
            ;4068 ; ------------------
            ;4069 ;
            ;4070 ;
            ;4071 ; Inputs:
            ;4072 ; ------
            ;4073 ;
            ;4074 ; NONE
            ;4075 ;
            ;4076 ;
            ;4077 ; Outputs:
            ;4078 ; -------
            ;4079 ;
            ;4080 ; Time-Out counter in Mic-Mon is Reset
            ;4081 ;
            ;4082 ;
            ;4083 ; Side Effects:
            ;4084 ; ------------
            ;4085 ;
            ;4086 ; Contents of the D reg are lost
            ;4087 ;
            ;4088 ;
            ;4089 ;
            ;4090 ;--
            ;4091 ;*****************************************************************
U 12AE, 0818,0038,0180,0800,0000,105E  ;4092 DELAY: D_K[.8],J/CALLCON
            ;4093
```

```
;4094  .PAGE "TEST 1BB COLUMN/ROW SELECT ADDRESSING"
;4095 ;************************************************************
;4096 ;***
;4097 ;
;4098 ; Functional Description:
;4099 ; ----------------------
;4100 ;
;4101 ; This test is designed to detect addressing errors in the RAM Column and
;4102 ; Row Addressing lines, bits <17:02> of the SBI REC lines or bits <19:02>
;4103 ; if 256k RAM chips are present.  The test keeps running count through
;4104 ; each field (Column or Row) while keeping the other constant at 0.
;4105 ;
;4106 ; The test makes a first "sweep" through the RAS/CAS lines writing data
;4107 ; of zero.  The next sweep reads back the data previously written and
;4108 ; makes sure that it is intact.  If it is, a unique Reed-Muller error
;4109 ; correcting code is written to the location.  A last "sweep" of the
;4110 ; RAS/CAS address lines is made reading the encoded patterns, decoding
;4111 ; them and making sure they are equal to the expected.
;4112 ;
;4113 ;   [Subtest 1:]
;4114 ;
;4115 ; Row Address Select
;4116 ;
;4117 ;   [Subtest 2:]
;4118 ;
;4119 ; Column Address Select
;4120 ;
;4121 ;
;4122 ; Logic Description:
;4123 ; -----------------
;4124 ;  N/A
;4125 ;
;4126 ; Error Logout:
;4127 ; ------------
;4128 ;
;4129 ; See individual error reports.
;4130 ;
;4131 ; Sync Point Description:
;4132 ; ----------------------
;4133 ;  N/A
;4134 ;
;4135 ;============================================================
;4136 ;
;4137 ; Register Map:
;4138 ; ------------
;4139 ;
;4140 ;      RA,RB      Description:
;4141 ;      -----      -----------
;4142 ;
;4143 ;        D RAS/CAS Flag  1 = Testing CAS
;4144 ;      0 = Testing RAS
;4145 ;
;4146 ;
;4147 ;       RC        Description:
;4148 ;       --        -----------
```

```
            ;4149 ;
            ;4150 ;       C 64k/256k Flag  1 = 256K chips on array
            ;4151 ;     0 = 64k chips on array
            ;4152 ;
            ;4153 ;       8 Maximum Pass Count (this is dependent on whether there is 64k
            ;4154 ;  or 256k chips on the array)
            ;4155 ;
            ;4156 ; 6 Saved Pattern counter
            ;4157 ;
            ;4158 ;       7 Running Pass Count
            ;4159 ;
            ;4160 ;       4 Pattern Counter for Encode and Decode
            ;4161 ;
            ;4162 ; 1 Write Data Generated from Encode
            ;4163 ;
            ;4164 ;       0 Read Data Save
            ;4165 ;
            ;4166 ;--
            ;4167 ;*************************************************************
            ;4168 =0
U 110A. 0000.003D.0180.0800.0000.114F  ;4169 T.31: NEWTST   ; Signify start of new test
U 110B. 0000.003C.0180.0800.0000.1050  ;4170  NOP
U 1050. 0000.003D.0180.0800.0000.109A  ;4171 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the system
U 1051. 0000.003C.0180.0800.0000.12EE  ;4172  J/T.31.EXIT  ; No MS780-E on this system, exit test.
U 1052. 0000.003C.0180.0800.0000.1060  ;4173  nop
            ;4174 =
            ;4175 =00
            ;4176 RESTART.TEST.31:
U 1060. 0000.003D.0180.0800.0000.105C  ;4177  CALL[START.TEST] ; Configure the controllers
U 1061. 0000.003C.0180.0800.0000.12EE  ;4178  J/T.31.EXIT  ; No more MS780-E's on the system, exit
U 1062. 0000.003C.0180.0800.0000.1064  ;4179  NOP
            ;4180 =
U 1064. 0000.003D.0180.0800.0000.1178  ;4181 =00 CALL[64K.OR.256K] ; Find out what kind of memory arrays are
            ;4182      ; present (64K or 256K)
U 1065. 0000.003C.0180.0800.0000.1110  ;4183  J/T.31.MIX.ERROR ; CASE 1:  Return here when there is a mixture
            ;4184      ; of arrays on the controller or there were
            ;4185      ; no arrays found.
U 1066. 0000.003C.0180.0800.0000.12AF  ;4186  J/T.31.64K.CHIPS ; CASE 2:  Return here when there is 64k ram
            ;4187      ; array boards present (1 mega byte array)
U 1067. 0000.003C.0180.0800.0000.12B4  ;4188  J/T.31.256K.CHIPS ; CASE 3:  Return here when there is 256k ram
            ;4189      ; array boards present (4 mega byte array)
```

```
            ;4190  .PAGE
            ;4191 ;
            ;4192 ;////////////////////////////////////////////////////////////////////////////
            ;4193 ;
            ;4194 ; Got an error from 64K.OR.256K subroutine.  Either there are no arrays present
            ;4195 ; or there are a mixture of 64K and 256K arrays.  Call an error.
            ;4196 ;
            ;4197 =0
            ;4198 T.31.MIX.ERROR:
U 1110, 0000,003D,0980,0800,0084,7054  ;4199  ERROR2[.2]
U 1111, 0000,003C,0180,0800,0000,12AF  ;4200  NOP
            ;4201 ;
            ;4202 ;////////////////////////////////////////////////////////////////////////////
            ;4203 ;
            ;4204 ; ERROR LOGOUT:
            ;4205 ;
            ;4206 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4207 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4208 ;
            ;4209 ;////////////////////////////////////////////////////////////////////////////
            ;4210 ;
            ;4211 T.31.64K.CHIPS:
U 12AF, 0003,003C,0180,09E0,0000,12B0  ;4212  RC[0C]_0  ; Clear flag to indicate that we have 64K chips
            ;4213      ; on the array
U 12B0, 0000,003C,0180,0800,0084,72B1  ;4214  SC_K[.8]  ; Set-up to mask bit 8
U 12B1, 0803,001C,0180,0800,0000,12B2  ;4215  D_NOT.MASK  ; Assert bit 9 in d register to get x100
U 12B2, 0819,0000,0580,0800,0000,12B3  ;4216  D_D-K[.1]  ; Subtract 1 to allow for count of Zero
U 12B3, 0001,003C,0180,09C0,0000,12B9  ;4217  RC[08]_D,J/T.31.S1 ; Move x100 to RC[08] to set-up maximum address
            ;4218      ; we would have to reference.  Go to subtest 1
            ;4219 T.31.256K.CHIPS:
U 12B4, 0018,0038,05B0,09E0,0000,12B5  ;4220  RC[0C]_K[.1]  ; Set flag to indicate that 256K chips were
            ;4221      ; found on the array.
U 12B5, 0000,003C,D980,0800,0084,72B6  ;4222  SC_K[.9]  ; Set-up to mask bit 9
U 12B6, 0803,001C,0180,0800,0000,12B7  ;4223  D_NOT.MASK  ; Assert bit 9 in d register to get x200
U 12B7, 0819,0000,0580,0800,0000,12B8  ;4224  D_D-K[.1]  ; Subtract 1 to allow for count of Zero
U 12B8, 0001,003C,0180,09C0,0000,12B9  ;4225  RC[08]_D,J/T.31.S1 ; Move x200 to RC[08] to set-up maximum address
            ;4226      ; we would have to reference.
            ;4227      ; Goto subtest 1
            ;4228 ;
            ;4229 ;============================================================================
            ;4230 ;
            ;4231 ; Subtest 1
            ;4232 ;
            ;4233 T.31.S1:
U 12B9, 0003,003C,0180,0AE8,0000,1112  ;4234  R[0D]_0  ; Clear Flag to indicate that RAS lines are
            ;4235      ; under test
U 1112, 0000,003D,0180,0800,0000,1116  ;4236 =0 CALL[TEST.31.TST] ; Execute the test for RAS
            ;4237 ;
            ;4238 ;============================================================================
            ;4239 ;
            ;4240 ; Subtest 2
            ;4241 ;
            ;4242 T.31.S2:
U 1113, 0018,0038,0580,0AE8,0000,1114  ;4243  R[0D]_K[.1]  ; Set flag to indicate that CAS lines are
            ;4244      ; under test
```

```
U 1114, 0000,003D,0180,0800,0000,1116  ;4245 =0 CALL[TEST.31.TST] ; Execute the test for CAS
U 1115, 0000,003C,0180,0800,0000,1060  ;4246  J/RESTART.TEST.31 ; Restart test for next MS780-E
          ;4247 ;
          ;4248 ; Here starts the main body of the test.  It is called as a subroutine since
          ;4249 ; the code is, more or less, the same for RAS and CAS.
          ;4250 ;
          ;4251 =0
          ;4252 TEST.31.TST:
U 1116, 0000,003D,0180,0800,0000,116C  ;4253  SUBTEST   ; Increment the Subtest counter
U 1117, 0000,003C,0180,0800,0000,1118  ;4254  NOP
U 1118, 0000,003D,0180,0800,0000,1098  ;4255 =0 CALL[SBI.INIT]  ; Initialize SBI
U 1119, 0000,003C,0180,0800,0000,1068  ;4256  NOP
U 1068, 0000,003D,0180,0800,0000,1192  ;4257 =00 CHECK.INTERRUPT  ; Clear previous interrupts
U 1069, 0000,003C,0180,0800,0000,106A  ;4258  NOP
U 106A, 0000,003C,0180,0800,0000,106B  ;4259  NOP
U 106B, 0003,003C,0180,0800,1408,72BA  ;4260  STATE_0(A)  ; Clear the error counter
U 12BA, 0018,0038,0980,09B8,0000,12BB  ;4261  RC[07]_K[.2]  ; Clear the pass counter
U 12BB, 0018,0038,0580,09A0,0000,12BC  ;4262  RC[04]_K[.1]  ; Clear the pattern counter
          ;4263 T.31.WRITE.ZEROS:
U 12BC, 0810,0038,0180,0938,0000,12BD  ;4264  D_RC[07]  ; Move pass counter to the D register
U 12BD, 0011,0020,0180,0940,0010,12BE  ;4265  ALU_D.XOR.RC[08],CLK.UBCC
          ;4266     ; Compare pass counter with maximum number
          ;4267     ; of passes required for this size array.
U 12BE, 0000,013C,0180,0800,0000,111A  ;4268  Z?   ; Is pass count and Maximum pass count equal ?
U 111A, 0000,003C,0180,0800,0000,12BF  ;4269 =0 J/T.31.CONT.WRITE.ZEROS ; No, continue writing zero's.
U 111B, 0000,003C,0180,0800,0000,1122  ;4270  J/T.31.SEC.TWO  ; Yes, go to the next part of this test.
          ;4271  ;
          ;4272  ; Come here when writing zero's to init memory and the pass count is
          ;4273  ; no equal to the maximum passes required for this size array.
          ;4274  ;
          ;4275 T.31.CONT.WRITE.ZEROS:
U 12BF, 0000,003C,0180,0A68,0010,12C0  ;4276  ALU_R[0D],CLK.UBCC ; Move RAS/CAS flag to the ALU.
U 12C0, 0000,013C,0180,0800,0000,111C  ;4277  Z?   ; Are we doing RAS or CAS ?
U 111C, 0000,003C,0180,0800,0000,111E  ;4278 =0 J/T.31.DO.CAS.1  ; Doing CAS, call CAS.ADD.GEN
U 111D, 0000,003C,0180,0800,0000,1120  ;4279  J/T.31.DO.RAS.1  ; Doing RAS, call RAS.ADD.GEN
          ;4280  ;
          ;4281  ; Come here when doing CAS tests to generate address.
          ;4282  ;
          ;4283 =0
          ;4284 T.31.DO.CAS.1:
U 111E, 0000,003D,0180,0800,0000,1206  ;4285  CALL[CAS.ADD.GEN] ; Generate CAS Address
U 111F, 0000,003C,0180,0800,0000,12C1  ;4286  J/T.31.DO.WRITE.ZEROS ; Go write the zeros to the address generated
          ;4287  ;
          ;4288  ; Come here when doing RAS tests to generate address.
          ;4289  ;
          ;4290 =0
          ;4291 T.31.DO.RAS.1:
U 1120, 0000,003D,0180,0800,0000,11FE  ;4292  CALL[RAS.ADD.GEN] ; Generate RAS Address
U 1121, 0000,003C,0180,0800,0000,12C1  ;4293  NOP   ; Fall through to write zero's to address
          ;4294  ;
          ;4295  ; Come here to do the actual write of zero's to the generated address
          ;4296  ;
          ;4297 T.31.DO.WRITE.ZEROS:
U 12C1, 0001,003C,0180,0800,0200,12C2  ;4298  VA_D   ; Get address to VA register
U 12C2, 0F00,003C,0180,0800,0000,12C3  ;4299  D_0   ; Zero the D register for writing zero's
```

```
ZZ-ESKAR-V22  TEST 1BB COLUMN/ROW SELECT ADDRESSING                                    SEQ 1229
ESKAR4A.MCR                    MICRO2  1P(00)    26-JUL-85  17:26:47                  Page  11

U 12C3, 0000,003C,0180,A800,0000,1080  ;4300  CACHE.P_D[LONG]  ; Write the zero's to the test address
U 1080, 0000,003D,0180,0800,0000,1192  ;4301 =00 CHECK.INTERRUPT  ; Were there any unexpected interrupts ?
U 1081, 0000,003C,0180,0800,0000,1083  ;4302  J/T.31.INC.PASS.COUNT ; No, Increment pass count and continue test
```

```
            ;4303  .PAGE
            ;4304 ;
            ;4305 ;////////////////////////////////////////////////////////////////////////////////
            ;4306 ;
            ;4307 ; Got an unexpected interrupt when writing zero's to the test address.
            ;4308 ; Call an error.
            ;4309 ;
U 1082, 0000,003D,8580,0800,0084,7054  ;4310  ERROR2[.C]
            ;4311 ;
            ;4312 ;////////////////////////////////////////////////////////////////////////////////
            ;4313 ;
            ;4314 ; ERROR LOGOUT:
            ;4315 ;
            ;4316 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4317 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4318 ;  Unused
            ;4319 ;  Unused
            ;4320 ;  Unused
            ;4321 ;  Unused
            ;4322 ;  ID[vector] Register
            ;4323 ;  ID[sbi.err] Register
            ;4324 ;  ID[fault] Register
            ;4325 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4326 ;      0 = No Interrupt Occurred
            ;4327 ;  Pattern Counter for Encode and Decode
            ;4328 ;  Unused
            ;4329 ;
            ;4330 ;////////////////////////////////////////////////////////////////////////////////
            ;4331 ;
            ;4332 T.31.INC.PASS.COUNT:
U 1083, 0810,0038,0180,0938,0000,12C4  ;4333  D_RC[07]  ; Move pass count to the D register
U 12C4, 0019,0014,0580,09B8,0000,12C5  ;4334  RC[07]_D+K[.1]  ; Increment the pass count
U 12C5, 0810,0038,0180,0920,0000,12C6  ;4335  D_RC[04]  ; Move Pattern count to the D register
            ;4336  RC[04]_D+K[.1],  ; Increment the Pattern count and return
U 12C6, 0019,0014,0580,09A0,0000,12BC  ;4337    J/T.31.WRITE.ZEROS ; begining of this section.
            ;4338  ;
            ;4339  ; This is section two of the test and here we read the locations that
            ;4340  ; were just wrote to see if the zero's got wrote with no problems.
            ;4341  ;
            ;4342 =0
            ;4343 T.31.SEC.TWO:
U 1122, 0000,003D,0180,0800,0000,12AE  ;4344  DELAY   ; Reset the time-out coutner in MICMON
U 1123, 0018,0038,0980,09B8,0000,12C7  ;4345  RC[07]_K[.2]  ; Clear Pass counter
U 12C7, 0018,0038,0580,09A0,0000,12C8  ;4346  RC[04]_K[.1]  ; Initialize the Pattern Counter
            ;4347 T.31.READ.ZEROS:
U 12C8, 0810,0038,0180,0938,0000,12C9  ;4348  D_RC[07]  ; Move pass counter to the D register
U 12C9, 0011,0020,0180,0940,0010,12CA  ;4349  ALU_D.XOR.RC[08],CLK.UBCC
            ;4350      ; Compare pass counter with maximum number
            ;4351      ; of passes required for this size array.
U 12CA, 0000,013C,0180,0800,0000,1124  ;4352  Z?   ; Is pass count and Maximum pass count equal ?
U 1124, 0000,003C,0180,0800,0000,12CB  ;4353 =0 J/T.31.CONT.READ.ZEROS ; No, continue writing zero's.
U 1125, 0000,003C,0180,0800,0000,1136  ;4354  J/T.31.SEC.THREE ; Yes, go to the next part of this test.
            ;4355  ;
            ;4356  ; Come here when reading zero's to check memory and the pass count is
            ;4357  ; no equal to the maximum passes required for this size array.
```

```
            ;4358  ;
            ;4359 T.31.CONT.READ.ZEROS:
U 12CB. 0000,003C,0180,0A68,0010,12CC  ;4360  ALU_R[0D],CLK.UBCC ; Move RAS/CAS flag to the ALU.
U 12CC. 0000,013C,0180,0800,0000,1126  ;4361  Z?   ; Are we doing RAS or CAS ?
U 1126. 0000,003C,0180,0800,0000,1128  ;4362 =0 J/T.31.DO.CAS.2  ; Doing CAS, call CAS.ADD.GEN
U 1127. 0000,003C,0180,0800,0000,112A  ;4363  J/T.31.DO.RAS.2  ; Doing RAS, call RAS.ADD.GEN
            ;4364  ;
            ;4365  ; Come here when doing CAS tests to generate address.
            ;4366  ;
            ;4367 =0
            ;4368 T.31.DO.CAS.2:
U 1128. 0000,003D,0180,0800,0000,1206  ;4369  CALL[CAS.ADD.GEN] ; Generate CAS Address
U 1129. 0000,003C,0180,0800,0000,12CD  ;4370  J/T.31.DO.READ.ZEROS ; Go read the zeros to the address generated
            ;4371  ;
            ;4372  ; Come here when doing RAS tests to generate address.
            ;4373  ;
            ;4374 =0
            ;4375 T.31.DO.RAS.2:
U 112A. 0000,003D,0180,0800,0000,11FE  ;4376  CALL[RAS.ADD.GEN] ; Generate RAS Address
U 112B. 0000,003C,0180,0800,0000,12CD  ;4377  NOP   ; Fall through to read zero's from address
            ;4378  ;
            ;4379  ; Come here to start the read from the array to check the zero's that
            ;4380  ; wrote in the previous section.
            ;4381  ;
            ;4382 T.31.DO.READ.ZEROS:
U 12CD. 0001,003C,0180,0800,0200,12CE  ;4383  VA_D   ; Move test address to the VA
U 12CE. 0001,003C,0180,09D8,0000,12CF  ;4384  RC[0B]_D  ; Save VA in RC[0B]
U 12CF. 0003,003C,0180,0AF8,0000,112C  ;4385  R[0F]_0   ; Clear Utrap flag
U 112C. 0000,003D,8980,0800,0084,719F  ;4386 =0 SET.TRAP.MASK[.D] ; Enable Time-Out micro traps
U 112D. 0000,003C,0180,C800,0000,12D0  ;4387  D[LONG]_CACHE.P  ; Read the test location pointed to by VA
U 12D0. 0001,003C,0180,0980,0000,12D1  ;4388  RC[00]_D  ; Save read data in RC[00]
            ;4389  ALU_R[0F],  ; Check to see if a Time-Out occured
U 12D1. 0000,003C,0180,0A78,0010,12D2  ;4390   CLK.UBCC ; Clock Condition codes
U 12D2. 0000,013C,0180,0800,0000,112E  ;4391  Z?   ; Is it zero ?
U 112E. 0000,003C,0180,0800,0000,1084  ;4392 =0 J/T.31.CHK.INT.1 ; No, go check for any unexpected interrupts
```

```
            ;4393  .PAGE
            ;4394 ;
            ;4395 ;////////////////////////////////////////////////////////////////////////////////
            ;4396 ;
            ;4397 ; Got a micro trap on the Read of the test location.  Call an error.
            ;4398 ;
U 112F, 0000,003D,0180,0800,0084,7054  ;4399  ERROR2[.8]
U 12D3, 0000,003C,0180,0800,0000,1084  ;4400  NOP
            ;4401 ;
            ;4402 ;////////////////////////////////////////////////////////////////////////////////
            ;4403 ;
            ;4404 ; ERROR LOGOUT:
            ;4405 ;
            ;4406 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4407 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4408 ;  Unused
            ;4409 ;  Unused
            ;4410 ;  Unused
            ;4411 ;  Unused
            ;4412 ;  Expected Micro Trap Flags
            ;4413 ;  Received Micro Trap Flags
            ;4414 ;
            ;4415 ;////////////////////////////////////////////////////////////////////////////////
            ;4416 ;
            ;4417 =00
            ;4418 T.31.CHK.INT.1:
U 1084, 0000,003D,0180,0800,0000,1192  ;4419  CHECK.INTERRUPT  ; Were there any unexpected interrupts ?
U 1085, 0000,003C,0180,0800,0000,12D4  ;4420  J/T.31.DO.DECODE.1 ; No, go do the decode
```

```
            ;4421  .PAGE
            ;4422 ;
            ;4423 ;////////////////////////////////////////////////////////////////////////////////
            ;4424 ;
            ;4425 ; Got an unexpected interrupt on the Read of the test location.  Call an error.
            ;4426 ;
U 1086, 0000,003D,8580,0800,0084,7054  ;4427  ERROR2[.C]
U 1087, 0000,003C,0180,0800,0000,12D4  ;4428  NOP
            ;4429 ;
            ;4430 ;////////////////////////////////////////////////////////////////////////////////
            ;4431 ;
            ;4432 ; ERROR LOGOUT:
            ;4433 ;
            ;4434 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4435 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4436 ;  Unused
            ;4437 ;  Unused
            ;4438 ;  Unused
            ;4439 ;  Unused
            ;4440 ;  ID[vector] Register
            ;4441 ;  ID[sbi.err] Register
            ;4442 ;  ID[fault] Register
            ;4443 ;  Interrupt Occurred Flag  1 = Interrupt Occurred
            ;4444 ;       0 = No Interrupt Occurred
            ;4445 ;  Pattern Counter for Encode and Decode
            ;4446 ;  Unused
            ;4447 ;
            ;4448 ;////////////////////////////////////////////////////////////////////////////////
            ;4449 ;
            ;4450 T.31.DO.DECODE.1:
U 12D4, 0810,0038,0180,0920,0000,12D5  ;4451  D_RC[04]  ; Move the pattern count to RC[04]
U 12D5, 0001,003C,0180,09B0,0000,12D6  ;4452  rc[06]_D  ; Save the pattern count in rc[06]
U 12D6, 0003,003C,0180,09A0,0000,1088  ;4453  RC[04]_0  ; Set pattern  count to zero
U 1088, 0000,003D,0180,0800,0000,1233  ;4454 =00 CALL[DECODE]  ; Check the integrity of the data
U 1089, 0000,003C,0180,0800,0000,1130  ;4455  J/T.31.CALL.ERDAT.1 ; Return1 if maximum errors
U 108A, 0000,003C,0180,0800,0000,108B  ;4456  NOP   ; Return2 if errors but not maximum
U 108B, 0000,003C,0180,0800,0000,1133  ;4457  J/T.31.DO.ENCODE.1 ; No errors or error count not exceeded
```

```
            ;4458  .PAGE
            ;4459 ;
            ;4460 ;////////////////////////////////////////////////////////////////////////////////
            ;4461 ;
            ;4462 ; Got to many data integrity errors from the array.  Call an error
            ;4463 ;
            ;4464 =0
            ;4465 T.31.CALL.ERDAT.1:
U 1130, 0000.003D.0180.0800.0000.11E9  ;4466  CALL[ERDAT]  ; To many errors, log the error
U 1131, 0000.003C.0180.0800.0000.1132  ;4467  NOP
U 1132, 0000.003D.6180.0800.0084.7054  ;4468 =0 ERROR2[.F]
            ;4469 ;
            ;4470 ;////////////////////////////////////////////////////////////////////////////////
            ;4471 ;
            ;4472 ; ERROR LOGOUT:
            ;4473 ;
            ;4474 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4475 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4476 ;  Received Memory Data
            ;4477 ;  Expected Memory Data
            ;4478 ;  Received Memory Data
            ;4479 ;  Expected Memory Data
            ;4480 ;  Received Memory Data
            ;4481 ;  Expected Memory Data
            ;4482 ;  Received Memory Data
            ;4483 ;  Expected Memory Data
            ;4484 ;  Received Memory Data
            ;4485 ;  Expected Memory Data
            ;4486 ;  Received Memory Data
            ;4487 ;  Expected Memory Data
            ;4488 ;  FFFFFFFF
            ;4489 ;
            ;4490 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4491 ;       terminated by a xFFFFFFFF.  If there are more errors then can
            ;4492 ;       be output on the error then the last longword wrote will be the
            ;4493 ;       xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4494 ;
            ;4495 ;////////////////////////////////////////////////////////////////////////////////
            ;4496 ;
            ;4497 T.31.DO.ENCODE.1:
U 1133, 0810.0038.0180.0930.0000.12D7  ;4498  D_rc[06]  ; Move saved pattern counter to the D register
U 12D7, 0001.003C.0180.09A0.0000.1134  ;4499  RC[04]_D  ; Restore pattern count in RC[04]
U 1134, 0000.003D.0180.0800.0000.120F  ;4500 =0 CALL[ENCODE]  ; Encode the pattern
U 1135, 0810.0038.0180.0908.0000.12D8  ;4501  D_RC[01]  ; Move encoded pattern to the D register
U 12D8, 0010.0038.0180.0958.0200.12D9  ;4502  VA_RC[0B]  ; Reload the VA for the write
U 12D9, 0000.003C.0180.A800.0000.12DA  ;4503  CACHE.P_D[LONG]  ; Write the encoded pattern
U 12DA, 0810.0038.0180.0920.0000.12DB  ;4504  D_RC[04]  ; Move pattern count to the D register
U 12DB, 0019.0014.0580.09A0.0000.12DC  ;4505  RC[04]_D+K[.1]  ; Point to the next pattern
U 12DC, 0310.0038.0180.0938.0000.12DD  ;4506  D_RC[07]  ; Move pass count to the D register
U 12DD, 0019.0014.0580.09B8.0000.12C8  ;4507  RC[07]_D+K[.1],J/T.31.READ.ZEROS
            ;4508      ; Increment the pass count and continue with
            ;4509      ; this section.
            ;4510  ;
            ;4511  ; This section of the test will read and decode the encoded data that
            ;4512  ; was written to the test addresses in the last section.
```

```
            ;4513  ;
            ;4514 =0
            ;4515 T.31.SEC.THREE:
U 1136, 0000,003D,0180,0800,0000,12AE  ;4516  DELAY   ; Reset the Time-out counter in MICMON
U 1137, 0018,0038,0980,09B8,0000,12DE  ;4517  RC[07]_K[.2]  ; Clear the pass counter
U 12DE, 0018,0038,0580,09A0,0000,12DF  ;4518  RC[04]_K[.1]  ; Initialize the Pattern counter
            ;4519 T.31.DO.DECODE.2:
U 12DF, 0810,0038,0180,0938,0000,12E0  ;4520  D_RC[07]  ; Move pass counter to the D register
U 12E0, 0011,0020,0180,0940,0010,12E1  ;4521  ALU_D.XOR.RC[08],CLK.UBCC
            ;4522      ; Compare pass counter with maximum number
            ;4523      ; of passes required for this size array.
U 12E1, 0000,013C,0180,0800,0000,1138  ;4524  Z?    ; Is pass count and Maximum pass count equal ?
U 1138, 0000,003C,0180,0800,0000,12E2  ;4525 =0 J/T.31.CONT.DECODE ; No, continue decoding
U 1139, 0000,003C,0180,0800,0000,12EA  ;4526  J/T.31.SEC.FOUR  ; Yes, go to the next part of this test.
            ;4527  ;
            ;4528  ; Come here when the memory test location and the pass count is
            ;4529  ; no equal to the maximum passes required for this size array.
            ;4530  ;
            ;4531 T.31.CONT.DECODE:
U 12E2, 0000,003C,0180,0A68,0010,12E3  ;4532  ALU_R[0D],CLK.UBCC ; Move RAS/CAS flag to the ALU.
U 12E3, 0000,013C,0180,0800,0000,113A  ;4533  Z?    ; Are we doing RAS or CAS ?
U 113A, 0000,003C,0180,0800,0000,113C  ;4534 =0 J/T.31.DO.CAS.3  ; Doing CAS, call CAS.ADD.GEN
U 113B, 0000,003C,0180,0800,0000,113E  ;4535  J/T.31.DO.RAS.3  ; Doing RAS, call RAS.ADD.GEN
            ;4536  ;
            ;4537  ; Come here when doing CAS tests to generate address.
            ;4538  ;
            ;4539 =0
            ;4540 T.31.DO.CAS.3:
U 113C, 0000,003D,0180,0800,0000,1206  ;4541  CALL[CAS.ADD.GEN] ; Generate CAS Address
U 113D, 0000,003C,0180,0800,0000,12E4  ;4542  J/T.31.DO.DECODE.3 ; Go decode the data in the address generated
            ;4543  ;
            ;4544  ; Come here when doing RAS tests to generate address.
            ;4545  ;
            ;4546 =0
            ;4547 T.31.DO.RAS.3:
U 113E, 0000,003D,0180,0800,0000,11FE  ;4548  CALL[RAS.ADD.GEN] ; Generate RAS Address
U 113F, 0000,003C,0180,0800,0000,12E4  ;4549  NOP   ; Fall through to decode data from address
            ;4550  ;
            ;4551  ; This part of this section will read the location under test and
            ;4552  ; then call decode to check the integrity.
            ;4553  ;
            ;4554 T.31.DO.DECODE.3:
U 12E4, 0001,003C,0180,0800,0200,12E5  ;4555  VA_D   ; Load VA with the address under test
U 12E5, 0000,003C,0180,C800,0000,12E6  ;4556  D[LONG]_CACHE.P  ; Read the test location
U 12E6, 0001,003C,0180,0980,0000,1090  ;4557  RC[00]_D  ; Save data read in RC[00]
U 1090, 0000,003D,0180,0800,0000,1233  ;4558 =00 CALL[DECODE]  ; Go decode the received data
U 1091, 0000,003C,0180,0800,0000,1140  ;4559  J/T.31.CALL.ERDAT.2 ; Return1 if maximum errors
U 1092, 0000,003C,0180,0800,0000,1093  ;4560  NOP   ; Return2 if errors but not maximum
U 1093, 0000,003C,0180,0800,0000,1143  ;4561  J/T.31.UPD.PATTERN ; Good data integrity, continue testing
```

```
            ;4562  .PAGE
            ;4563 ;
            ;4564 ;////////////////////////////////////////////////////////////////////////////////
            ;4565 ;
            ;4566 ; To many errors from memory during the decode.  Call an error.
            ;4567 ;
            ;4568 =0
            ;4569 T.31.CALL.ERDAT.2:
U 1140, 0000,003D,0180,0800,0000,11E9  ;4570  CALL[ERDAT]  ; Get the data for logout
U 1141, 0000,003C,0180,0800,0000,1142  ;4571  NOP
U 1142, 0000,003D,6180,0800,0084,7054  ;4572 =0 ERROR2[.F]
            ;4573 ;
            ;4574 ;////////////////////////////////////////////////////////////////////////////////
            ;4575 ;
            ;4576 ; ERROR LOGOUT:
            ;4577 ;
            ;4578 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4579 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4580 ;  Received Memory Data
            ;4581 ;  Expected Memory Data
            ;4582 ;  Received Memory Data
            ;4583 ;  Expected Memory Data
            ;4584 ;  Received Memory Data
            ;4585 ;  Expected Memory Data
            ;4586 ;  Received Memory Data
            ;4587 ;  Expected Memory Data
            ;4588 ;  Received Memory Data
            ;4589 ;  Expected Memory Data
            ;4590 ;  Received Memory Data
            ;4591 ;  Expected Memory Data
            ;4592 ;  FFFFFFFF
            ;4593 ;
            ;4594 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4595 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4596 ;        be output on the error then the last longword wrote will be the
            ;4597 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4598 ;
            ;4599 ;////////////////////////////////////////////////////////////////////////////////
            ;4600 ;
            ;4601 T.31.UPD.PATTERN:
U 1143, 0810,0038,0180,0938,0000,12E7  ;4602  D_RC[07]  ; Increment pass counter
U 12E7, 0019,0014,0580,09B8,0000,12E8  ;4603  RC[07]_D+K[.1]  ; ...
U 12E8, 0810,0038,0180,0920,0000,12E9  ;4604  D_RC[04]  ; Move pattern count to the D register
            ;4605  RC[04]_D+K[.1],  ; Update pattern count for next pattern
U 12E9, 0019,0014,0580,09A0,0000,12DF  ;4606  J/T.31.DO.DECODE.2 ; and continue testing.
            ;4607  ;
            ;4608  ; This is the last section of this test and here we check the value
            ;4609  ; of state.  IF state is not equal to zero then get data for logout
            ;4610  ; and call error1, Else return to the caller.
            ;4611  ;
            ;4612 T.31.SEC.FOUR:
U 12EA, 0000,003C,0180,0800,1480,12EB  ;4613  SC_STATE  ; Move STATE to the SC
U 12EB, 0018,0038,1DC0,0800,0000,12EC  ;4614  Q_SC  ; Put SC (STATE) in Q
U 12EC, 0001,203C,0180,0800,0010,12ED  ;4615  ALU_Q,CLK.UBCC  ; Move Q (State) into the ALU for check
U 12ED, 0000,013C,0180,0800,0000,1144  ;4616  Z?  ; Is state 0 ?
```

```
U 1144, 0000,003C,0180,0800,0000,1146  ;4617 =0 J/T.31.STATE.NE.ZERO ; No, get the data for logout and call an error
U 1145, 0000,003E,0180,0800,0000,0001  ;4618  RETURN1   ; Yes, return to the caller
            ;4619  ;
            ;4620  ; Come here after checking the value of state and finding that it is
            ;4621  ; non zero.  Get the data for the logout and call an error.
            ;4622  ;
            ;4623 =0
            ;4624 T.31.STATE.NE.ZERO:
U 1146, 0000,003D,0180,0800,0000,11E9  ;4625  CALL[ERDAT]  ; Get the data for logout
U 1147, 0000,003C,0180,0800,0000,1148  ;4626  NOP
```

```
            ;4627   .PAGE
            ;4628 ;
            ;4629 ;////////////////////////////////////////////////////////////////////////////////
            ;4630 ;
            ;4631 ; This will call an error from state no equal to zero.
            ;4632 ;
U 1148. 0000.003D.6180.0800.0084.7054  ;4633 =0 ERROR2[.F]
U 1149. 0000.003E.0180.0800.0000.0001  ;4634  RETURN1
            ;4635 ;
            ;4636 ;////////////////////////////////////////////////////////////////////////////////
            ;4637 ;
            ;4638 ; ERROR LOGOUT:
            ;4639 ;
            ;4640 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4641 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4642 ;  Received Memory Data
            ;4643 ;  Expected Memory Data
            ;4644 ;  Received Memory Data
            ;4645 ;  Expected Memory Data
            ;4646 ;  Received Memory Data
            ;4647 ;  Expected Memory Data
            ;4648 ;  Received Memory Data
            ;4649 ;  Expected Memory Data
            ;4650 ;  Received Memory Data
            ;4651 ;  Expected Memory Data
            ;4652 ;  Received Memory Data
            ;4653 ;  Expected Memory Data
            ;4654 ;  FFFFFFFF
            ;4655 ;
            ;4656 ; NOTE:  The combination "Received and Expected Memory Data" will be
            ;4657 ;        terminated by a xFFFFFFFF.  If there are more errors then can
            ;4658 ;        be output on the error then the last longword wrote will be the
            ;4659 ;        xFFFFFFFF, if not then XFFFFFFFF will appear earlier.
            ;4660 ;
            ;4661 ;////////////////////////////////////////////////////////////////////////////////
            ;4662 ;
            ;4663  ;
            ;4664  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4665  ;
            ;4666 T.31.EXIT:
U 12EE. 0000.003C.0180.0800.0000.114A  ;4667  NOP
            ;4668
            ;4669
```

```
            ;4670  .PAGE "TEST 1BC RESERVED"
            ;4671 =0
U 114A, 0000,003D,0180,0800,0000,114F  ;4672 T1BC: NEWTST
U 114B, 0000,003C,0180,0800,0000,114C  ;4673  NOP
            ;4674
```

```
            ;4675  .PAGE "DUMMY NEWTST"
            ;4676 =0
U 114C, 0000,003D,0180,0800,0000,114F  ;4677 DUM: NEWTST
            ;4678
```

ZZ-ESKAR-V22 Line Number Index
ESKAR4A.MCR MICRO2 1P(00) 26-JUL-85 17:26:47
: Location / Line Number Index

SEQ 1241
Page 12

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1931: 1910 1919= 1920= 2766= 2767= 2768= 2771=
U 1008  2422= 2424= 2426= 2431= 2272= 2273= 2274= 2275=
U 1010  2445= 2447= 2448= 2452= 2468= 2469= 2470= 2478=
U 1018  2480= 2482= 2484= 2485= 2411= 2412= 2413= 2414=
U 1020  2529= 2530= 2531= 2533= 2577= 2578= 2579= 2581=
U 1028  2863= 2864= 2865= 2866= 2867= 2868= 2869= 2870=
U 1030  3103= 3104= 3111= 3113= 4025= 4026= 4027= 4029=
U 1038  1921= 1922= 1911 3143= 1974= 1976= 1912 3144=
U 1040  4031= 4032= 4035= 4038= 1987= 1988= 1913 3780=
U 1048  4043= 4045= 4047= 4050= 2048= 2049= 2030: 3784=
U 1050  4171= 4172= 4173= 1915 2094= 2095= 1916 3875=
U 1058  2219= 2220= 2076: 1923 2400= 2401= 2110: 3878=
U 1060  4177= 4178= 4179= 1924 4181= 4183= 4186= 4188=
U 1068  4257= 4258= 4259= 4260= 2403= 2408= 3229= 3230=
U 1070  3466= 3467= 3468= 3469= 3470= 3471= 3472= 3473=
U 1078  3474= 3475= 3476= 3477= 3478= 3479= 3480= 3483=
U 1080  4301= 4302= 4310= 4333= 4419= 4420= 4427= 4428=
U 1088  4454= 4455= 4456= 4457= 2612= 2617= 3766= 3767=
U 1090  4558= 4559= 4560= 4561= 2622= 2627= 2632= 2633=
U 1098  2678= 2679= 2842= 2843= 2851= 2852= 3862= 3863=
U 10A0  2853= 2855= 2858= 2859= 2900= 2901= 1925 3927=
U 10A8  2902= 2903= 2910= 2911= 2917= 2919= 1926 3928=
U 10B0  2922= 2923= 2942= 2943= 2945= 2946= 3037= 3038=
U 10B8  3041= 3042= 3053= 3054= 3057= 3058= 3288= 3290=
U 10C0  3347= 3348= 3417= 3419= 3422= 3424= 3425= 3427=
U 10C8  3539= 3540= 3544= 3545= 3559= 3561= 3619= 3620=
U 10D0  3624= 3625= 3627= 3628= 3639= 3640= 3652= 3653=
U 10D8  3716= 3718= 3725= 3726= 3733= 3734= 3741= 3742=
U 10E0  3747= 3750= 3774= 3776= 3789= 3793= 3796= 3797=
U 10E8  3803= 3806= 3828= 3830= 3837= 3838= 3843= 3846=
U 10F0  3870= 3872= 3882= 3883= 3885= 3888= 3901= 3903=
U 10F8  3912= 3915= 3920= 3921= 3944= 3946= 3967= 3969=
U 1100  1896: 1897: 1898: 1899: 1900: 1901: 1902: 1903:
U 1108  1904: 1927 4169= 4170= 1905: 1906: 1907: 1908:
U 1110  4199= 4200= 4236= 4243= 4245= 4246= 4253= 4254=
U 1118  4255= 4256= 4269= 4270= 4278= 4279= 4285= 4286=
U 1120  4292= 4293= 4344= 4345= 4353= 4354= 4362= 4363=
U 1128  4369= 4370= 4376= 4377= 4386= 4387= 4392= 4399=
U 1130  4466= 4467= 4468= 4498= 4500= 4501= 4516= 4517=
U 1138  4525= 4526= 4534= 4535= 4541= 4542= 4548= 4549=
U 1140  4570= 4571= 4572= 4602= 4617= 4618= 4625= 4626=
U 1148  4633= 4634= 4672= 4673= 4677= 1928 1929 1963
U 1150  1964 1965 1966 1967 1968 1969 1970 1971
U 1158  1972 1973 1977 1978 1979 1980 1981 1982
U 1160  1983 1984 1985 1986 2005 2028 2029 2074
U 1168  2075 2158 2159 2160 2179 2181 2208 2209
U 1170  2210 2211 2212 2213 2214 2215 2217 2218
U 1178  2265 2266 2267 2269 2271 2409 2410 2435
U 1180  2455 2456 2457 2464 2607 2656 2680 2681
U 1188  2682 2683 2702 2703 2704 2705 2724 2725
U 1190  2726 2727 2755 2757 2758 2759 2761 2762
U 1198  2763 2764 2765 2772 2773 2774 2775 2802
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2803 2804 2844 2845 2846 2847 2856 2857
U 11A8  2861 2862 2875 2876 2878 2879 2880 2882
U 11B0  2887 2888 2889 2891 2896 2898 2899 2907
U 11B8  2908 2912 2913 2914 2915 2916 2924 2925
U 11C0  2926 2928 2929 2930 2931 2932 2933 2938
U 11C8  2940 2941 2944 2969 2970 2971 2972 3001
U 11D0  3002 3004 3005 3006 3033 3035 3036 3040
U 11D8  3044 3045 3046 3047 3049 3050 3051 3052
U 11E0  3055 3056 3097 3098 3099 3100 3101 3115
U 11E8  3117 3135 3136 3137 3138 3139 3140 3141
U 11F0  3142 3146 3170 3171 3172 3173 3174 3175
U 11F8  3223 3224 3225 3226 3227 3228 3277 3279
U 1200  3280 3282 3283 3284 3286 3287 3336 3337
U 1208  3339 3340 3341 3342 3344 3345 3352 3409
U 1210  3411 3412 3413 3415 3416 3461 3464 3532
U 1218  3533 3535 3537 3538 3547 3549 3550 3551
U 1220  3552 3554 3556 3611 3612 3613 3615 3617
U 1228  3618 3630 3631 3635 3637 3641 3643 3644
U 1230  3645 3647 3649 3694 3695 3696 3697 3698
U 1238  3700 3701 3702 3703 3704 3705 3706 3708
U 1240  3709 3711 3712 3714 3715 3719 3720 3721
U 1248  3723 3724 3727 3728 3731 3732 3735 3736
U 1250  3737 3739 3740 3744 3745 3752 3753 3755
U 1258  3758 3759 3760 3762 3763 3764 3765 3770
U 1260  3771 3779 3786 3788 3794 3798 3799 3800
U 1268  3801 3802 3807 3809 3810 3812 3813 3814
U 1270  3815 3816 3817 3818 3820 3821 3823 3824
U 1278  3826 3827 3831 3832 3833 3835 3836 3840
U 1280  3841 3848 3849 3851 3854 3855 3856 3858
U 1288  3859 3860 3861 3866 3867 3874 3880 3881
U 1290  3884 3889 3890 3892 3893 3895 3897 3899
U 1298  3905 3907 3910 3918 3919 3924 3926 3939
U 12A0  3941 3942 3949 3952 3955 3956 3958 3959
U 12A8  3960 3961 2136: 3962 3964 3966 4092 4212
U 12B0  4214 4215 4216 4217 4220 4222 4223 4224
U 12B8  4225 4234 4261 4262 4264 4265 4268 4276
U 12C0  4277 4298 4299 4300 4334 4335 4337 4346
U 12C8  4348 4349 4352 4360 4361 4383 4384 4385
U 12D0  4388 4390 4391 4400 4451 4452 4453 4499
U 12D8  4502 4503 4504 4505 4506 4507 4518 4520
U 12E0  4521 4524 4532 4533 4555 4556 4557 4603
U 12E8  4604 4606 4613 4614 4615 4616 4667
U 12F0 - 13EF Unused
U 13F0  2120: 2121: 2122: 2123: 2124: 2125: 2126: 2127:
U 13F8  2128: 2129: 2130: 2131: 2132: 2133: 2134: 2135:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR4A	767
Used	767
Remaining	

Total microwords used in memory U: 767
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4A.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 1245
ESKAR4B.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2173 Diagnostic Specific Subroutines
; 2174   Select Controller
; 2212   START TEST
; 2421   SELECT LOWER CONTROLLER
; 2469   SELECT UPPER CONTROLLER
; 2517   Set Diagnostic Mode Routine
; 2546   SBI Unjam Routine
; 2595   RESET SUBTEST NUMBER
; 2617   Initialize the SBI
; 2644   Disable Cache Routine
; 2665   Enable Cache
; 2686   Wait Loop Routine
; 2719   Check for Interrupt Routine
; 2766   CHECK UTRAP
; 2798   Set Trap Mask
; 2826   FIND MS780-E MEMORY
; 2968   Generate IO Address
; 2993   Set Starting Address
; 3026   ENABLE NEXT MS780-E
; 3078   Set ECC Bits
; 3108   MS780-E/H CNFG REG CLEANUP
; 3178   CHECK SBI.ERR
; 3209   Check Syndrome For Even Number of One's
; 3251 TEST 1BD CHECK BITS GENERATION
; 3446 TEST 1BE SINGLE BIT ERROR DETECTION
; 3826 TEST 1BF SINGLE BIT ERROR CORRECTION PART 1
; 4116 TEST 1C0 SINGLE BIT ERROR CORRECTION PART 2
; 4444 TEST 1C1 SINGLE BIT ERROR CORRECTION PART 3
; 4842 TEST 1C2 DOUBLE BIT ERROR TEST PART 1
; 5256 TEST 1C3 DOUBLE BIT ERROR PART 2
; 5645 TEST 1C4 RESERVED
; 5650 TEST 1C5 RESERVED
; 5655 TEST 1C6 RESERVED
; 5660 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR4B
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
;1841   .PAGE
;1842
;1843   .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844   .TOC "    Micro Trap Handler"
;1845  ;**
;1846  ; FUNCTIONAL DESCRIPTION:
;1847  ;
;1848  ; This routine is responsible for 'catching' micro-traps
;1849  ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850  ; contains the Trap-Expected Mask. If the specified bit is set in
;1851  ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852  ; an error call is made.
;1853  ;
;1854  ; INPUT PARAMETERS:
;1855  ;
;1856  ; R[0F] - Expected Trap Mask
;1857  ;  Trap Code Function
;1858  ;  ---------      --------
;1859  ;       0  System Init
;1860  ;       1  Data Unaligned
;1861  ;       2  Cross Page
;1862  ;       3  TB Modify
;1863  ;       4  TB Protection
;1864  ;       6  TB Miss
;1865  ;       7  TB Parity
;1866  ;       8  Cache Parity
;1867  ;       9  Reserved Floating Operand
;1868  ;       C  Read Data Substitute
;1869  ;       D  Time out
;1870  ;       F  Control Store Parity Error
;1871  ;      10  Odd Address
;1872  ;
;1873  ; OUTPUT PARAMETERS:
;1874  ;
;1875  ; R[0F] - Mask bit is cleared.
;1876  ;
;1877  ; DATA: Micro PC of Micro Trap
;1878  ;  Trap Code (See Above)
;1879  ;  Fault Status Register
;1880  ;  SBI Error Register
;1881  ;  Timeout Address Register
;1882  ;  Maintenance Register
;1883  ;  Cache Parity Register
;1884  ;  TB Error Register 1
;1885  ;  TB Error Register 0
;1886  ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000.003C.8580.0800.0084.7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000.003C.8980.0800.0084.7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000.003C.6580.0800.0084.7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000.003C.6180.0800.0084.7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1900
U 1001, 0000.003C.81F0.2C00.0000.1033  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 1033, 0C00.003C.01E0.0800.0000.103B  ;1902  q_d,d_q   ; Setup to put it back on stack
U 103B, 0C00.003C.81E0.3C00.0000.1046  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1046, 0000.003C.01C0.0A78.0000.104B  ;1904  q_r[0f]   ; Get the Expected Trap Mask
          ;1905  alu_q.andnot.mask,
U 104B, 0001.2024.0180.0800.0010.1053  ;1906   clk.ubcc ; Was trap expected?
U 1053, 0000.013C.0180.0800.0000.1002  ;1907  z?
          ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1909       alu_q.and.mask, ; ...
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1910       return0  ; ...
U 1003, 0018.0038.1D80.09E8.0000.102C  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 102C, 0000.003D.D980.0800.0084.7231  ;1912 =0 setrcf[.9]  ; Set output count
U 102D, 0000.003C.49F0.2C00.0000.1056  ;1913  q_id[tber0]  ; Output all the error registers
U 1056, 0001.203C.4DF0.2DB0.0000.105B  ;1914  rc[6]_q,q_id[tber1] ; ...
U 105B, 0001.203C.65F0.2DB8.0000.1063  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 1063, 0001.203C.6DF0.2DD8.0000.1066  ;1916  rc[0b]_q,q_id[fault] ; ...
U 1066, 0001.203C.75F0.2DE0.0000.106E  ;1917  rc[0c]_q,q_id[maint] ; ...
U 106E, 0001.203C.79F0.2DC8.0000.1073  ;1918  rc[9]_q,q_id[parity] ; ...
U 1073, 0001.203C.69F0.2DC0.0000.1077  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 1077, 0001.203C.81F0.2DD0.0000.1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
          ;1921 1000: ERROR1[.C],  ;
U 1000, 0001.203D.8580.09F0.0084.704C  ;1922   rc[0e]_q ; Report the error
```

```
            ;1923  .PAGE
            ;1924  .TOC " Micro Monitor Interface Routines"
            ;1925  .TOC "    Newtest Routine"
            ;1926 ;++
            ;1927 ; FUNCTIONAL DESCRIPTION:
            ;1928 ;
            ;1929 ; This routine Initializes the following registers:
            ;1930 ;  PSL    to 1F
            ;1931 ;  CES    to 0
            ;1932 ;  ACC.CS    to disable accellerator
            ;1933 ;  SIR    to 0
            ;1934 ;  CLK.CS    to stop interval timer
            ;1935 ;  TBER0    to disable the TB
            ;1936 ;  SBI.MAINT to     0
            ;1937 ;  SBI.ERR    to 0
            ;1938 ;  FAULT    to 0
            ;1939 ;  COMP    to 0
            ;1940 ;  RC[0E]    to  0
            ;1941 ;  R[0F]    to 0
            ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1943 ; A SCOPE call is then made to the Monitor.
            ;1944 ;
            ;1945 ; CALLING CONSTRAINT:
            ;1946 ;
            ;1947 ; =0
            ;1948 ;
            ;1949 ; SIDE EFFECTS:
            ;1950 ;
            ;1951 ; D Reg is destroyed
            ;1952 ; SC is destroyed
            ;1953 ;--
U 107F, 0000.003C.65F8.0800.0084.7083  ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1083, 0818.0038.8D80.0800.0000.1087  ;1955  d_k[.1f]  ; D=1F
U 1087, 0D00.003C.0180.0800.0000.108B  ;1956  d_dai.sc  ; D=001F0000
U 108B, 0000.003C.3D80.3C00.0000.108F  ;1957  id[psl]_d  ; psl = 001F0000
U 108F, 0818.0038.0580.0800.0000.1093  ;1958  d_k[.1]  ; Clear the CES register
U 1093, 0D00.003C.0180.0800.0000.109B  ;1959  d_dai.sc  ; D = 00010000
U 109B, 0F00.003C.3180.3C00.0000.109F  ;1960  id[ces]_d,d_0  ; ces = 00010000
U 109F, 0000.003C.5D80.3C00.0000.10A3  ;1961  id[acc.cs]_d  ; acc.cs = 0
U 10A3, 0000.003C.3980.3C00.0000.10A7  ;1962  id[sir]_d  ; sir = 0
U 10A7, 0000.003C.2980.3C00.0000.10AB  ;1963  id[clk.cs]_d  ; clk.cs = 0
U 10AB, 0000.003C.4980.3C00.0000.103C  ;1964  id[tber0]_d  ; tber0 = 0
U 103C, 0000.003D.0180.0800.0000.124D  ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1966  intrpt.strobe,  ; Strobe interrupts
U 103D, 0818.0038.C180.0800.4000.10AF  ;1967      d_k[.ffff]  ; Setup to clear SBI error register and
U 10AF, 0801.0028.6580.3C00.0000.10B3  ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 10B3, 0F00.003C.6D80.3C00.0000.10B7  ;1969  id[fault]_d,d_0  ; ...
U 10B7, 0000.003C.6D80.3C00.0000.1109  ;1970  id[fault]_d  ; fault = 0
U 1109, 0000.003C.7580.3C00.0000.1226  ;1971  id[maint]_d  ; MAINT = 0
U 1226, 0000.003C.6580.3C00.0000.1227  ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 1227, 0000.003C.7180.3C00.0000.1228  ;1973  id[comp]_d  ; comp reg = 0
U 1228, 0000.003C.E580.3C00.0000.1229  ;1974  ID[39]_d  ; Clear code for subroutine START.TEST
U 1229, 0001.003C.0180.09F0.0000.122A  ;1975  rc[0e]_d  ;
U 122A, 0001.003C.0180.0AF8.0000.122B  ;1976  r[0f]_d  ; Clear expected trap mask
U 122B, 0001.003C.0180.09F8.0000.1044  ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1044, 0000,003D,0180,0800,0000,1253  ;1978 =0 call[disable.cache] ; Disable the cache
U 1045, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1980  .PAGE
            ;1981  .TOC "   Error Loop Routine"
            ;1982 ;++
            ;1983 ; FUNCTIONAL DESCRIPTION:
            ;1984 ;
            ;1985 ; This routine is called with the "ERLOOP" macro. The
            ;1986 ; routine calls the Micro Monitor to setup an error loop.
            ;1987 ;
            ;1988 ; CALLING CONSTRAINT:
            ;1989 ;
            ;1990 ; =0
            ;1991 ;
            ;1992 ; SIDE EFFECTS:
            ;1993 ;
            ;1994 ; D Reg - Destroyed
            ;1995 ;--
U 122C, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 122D, 0810,0038,0180,0978,0000,122E  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 122E, 0819,0034,4D80,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;++
            ;2025 ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 104C, 0000,003D,0180,0800,0000,1231  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 104D, 0000,003C,0180,0800,0000,122D  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "   Error 2 Routine"
            ;2045 ;**
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 122F, 0810,0038,0180,0978,0000,1230  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 1230, 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "   ERROR 2 WITH PARAMETER"
            ;2071 ;++
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 1054, 0000,003D,0180,0800,0000,1231  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1055, 0000,003C,0180,0800,0000,122F  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
            ;2088  .PAGE
            ;2089  .TOC "   Micro-Monitor Call"
            ;2090 ;+
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;++
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0, 0000,003C,0180,0800,0000,13F1   ;2111 13F0: nop
U 13F1, 0000,003C,0180,0800,0000,13F2   ;2112 13F1: nop
U 13F2, 0000,003C,0180,0800,0000,13F3   ;2113 13F2: nop
U 13F3, 0000,003C,0180,0800,0000,13F4   ;2114 13F3: nop
U 13F4, 0000,003C,0180,0800,0000,13F5   ;2115 13F4: nop
U 13F5, 0000,003C,0180,0800,0000,13F6   ;2116 13F5: nop
U 13F6, 0000,003C,0180,0800,0000,13F7   ;2117 13F6: nop
U 13F7, 0000,003C,0180,0800,0000,13F8   ;2118 13F7: nop
U 13F8, 0000,003C,0180,0800,0000,13F9   ;2119 13F8: nop
U 13F9, 0000,003C,0180,0800,0000,13FA   ;2120 13F9: nop
U 13FA, 0000,003C,0180,0800,0000,13FB   ;2121 13FA: nop
U 13FB, 0000,003C,0180,0800,0000,13FC   ;2122 13FB: nop
U 13FC, 0000,003C,0180,0800,0000,13FD   ;2123 13FC: nop
U 13FD, 0000,003C,0180,0800,0000,13FE   ;2124 13FD: nop
U 13FE, 0000,003C,0180,0800,0000,13FF   ;2125 13FE: nop
U 13FF, 0000,003C,0180,0800,0000,12AA   ;2126 13FF: nop
U 12AA, 0000,003C,0180,0800,0000,1231   ;2127 12AA: nop ; This location is permanently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;++
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 1231, 0010,0038,01C0,0978,0000,1232  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 1232, 0019,2034,4DC0,0800,0000,1233  ;2150  q_q.and.k[.ff00] ; ...
U 1233, 0019,2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "   Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 1234, 0810,0038,4D80,0978,0000,1235  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 1235, 0019,4016,4D80,09F8,0000,0001  ;2171      word,return1 ; (Subtest number is negative)
```

```
            ;2172  .PAGE
            ;2173  .TOC " Diagnostic Specific Subroutines"
            ;2174  .TOC "   Select Controller"
            ;2175 ;++
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine is used to put the memory system into the
            ;2179 ; specified configuration with respect to controller select.
            ;2180 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2181 ; a RETURN2 is made.
            ;2182 ;
            ;2183 ; CALLING CONSTRAINT:
            ;2184 ;
            ;2185 ; =00
            ;2186 ;
            ;2187 ; INPUTS:
            ;2188 ;
            ;2189 ; d - Contains value to write to bits<2:0> of Register A
            ;2190 ; rc[0e] - Address of Register A
            ;2191 ;
            ;2192 ; SIDE EFFECTS:
            ;2193 ;
            ;2194 ; sc,d,va are destroyed
            ;2195 ;--
            ;2196 SELECT.CNTRLR:
U 1236, 0010.003B.0180.0970.0284.7237  ;2197  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1237, 0801.001C.0180.0800.0000.1238  ;2198  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1238, 0000.003C.0180.A800.0000.1239  ;2199  cache.p_d[long]  ; Write the configuration Register
U 1239, 0818.003B.5D80.0800.0000.123A  ;2200  d_k[.7]  ; Get Misconfiguration bits
U 123A, 0000.003C.61F8.0800.0084.723B  ;2201  sc_k[.F],q_0  ; ...
U 123B, 0D00.003C.0180.0800.0000.123C  ;2202  d_dal.sc  ; ... D = x38000
U 123C, 0000.003C.01E0.0800.0000.123D  ;2203  q_d  ; Save mask
U 123D, 0000.003C.0180.C800.0000.123E  ;2204  d[long]_cache.p  ; Read CNFG A Reg and
            ;2205  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 123E, 001D.2034.0180.0800.0010.123F  ;2206  clk.ubcc ; ...
U 123F, 0000.013C.0180.0800.0000.1058  ;2207  z?  ; Branch if no
U 1058, 0000.003E.0180.0800.0000.0001  ;2208 =0 RETURN1  ; Bad configuration
U 1059, 0000.003E.0180.0800.0000.0002  ;2209  RETURN2  ; Configuration ok
            ;2210
```

```
;2211  .PAGE
;2212  .TOC "   START TEST"
;2213 ;*************************************************************
;2214 ;**
;2215 ;
;2216 ; Functional Description:
;2217 ; ---------------------
;2218 ;
;2219 ;        This subroutine will be called by all tests that  check  the
;2220 ;    controllers,  or  those  tests  that  have  to switch between the
;2221 ;    controllers when executing.  Its  main  functions  are:  select
;2222 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2223 ;    test that called it, call out an error  if  neither  one  of  the
;2224 ;    controllers can be configured properly, if all the controllers on
;2225 ;    a MS780-E/H were configured and tested, it should select the next
;2226 ;    MS780-E/H on the SBI and repeat the above sequence.
;2227 ;    A test making use of this  subroutine  should  be  configured  as
;2228 ;    follows:
;2229 ;
;2230 ;
;2231 ;       Begin TEST.XX
;2232 ;        :
;2233 ;        :   Call NEWTST
;2234 ;        :   Call FIND.MS780E
;2235 ;        :   IF No MS780-E's on the SBI
;2236 ;        :     THEN
;2237 ;        :         Skip to End of TEST.XX
;2238 ;        :     ELSE
;2239 ;        :
;2240 ;        : RESTART.TEST.XX:
;2241 ;        :
;2242 ;        :   Call START TEST
;2243 ;        :   IF Return1 from START.TEST
;2244 ;        :     THEN
;2245 ;        :         Skip to End of TEST.XX
;2246 ;        :     ELSE.
;2247 ;        :
;2248 ;        -
;2249 ;
;2250 ;        :
;2251 ;        :   Body of TEST.XX
;2252 ;        :
;2253 ;        -
;2254 ;
;2255 ;        :   Jump RESTART.TEST.XX
;2256 ;        :
;2257 ;       End TEST.XX
;2258 ;
;2259 ;
;2260 ;        This subroutine makes use of a pointer  in  ID  Register  39
;2261 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2262 ;    should be passed when the subroutine is called a second, third or
;2263 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2264 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2265 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2266 ;    Register 39 can be interpreted as follows:
;2267 ;
;2268 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2269 ;               called the first time.  When this code is encountered
;2270 ;               this subroutine will try to select the lower controller.
;2271 ;               If the lower controller is misconfigured, the pointer in
;2272 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2273 ;               is re-entered.
;2274 ;               If, however there is no misconfiguration, the value in
;2275 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2276 ;               calling test.
;2277 ;
;2278 ;     b01   -   This value signifies that an attempt at configuring the
;2279 ;               lower controller failed, and the subroutine will attempt
;2280 ;               to select the upper controller.
;2281 ;               If the upper controller is also misconfigured execution
;2282 ;               stops with a call to ERROR1.
;2283 ;               If there is no misconfiguration on this controller, then
;2284 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2285 ;               made to the calling test.
;2286 ;
;2287 ;     b10   -   This value signifies that the lower controller was selected
;2288 ;               previously and the test was executed once.  If the
;2289 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2290 ;               39 is loaded with b11 and an attempt is made to select the
;2291 ;               upper controller.
;2292 ;               If there is a misconfiguration on this controller, this
;2293 ;               subroutine is just re-entered.
;2294 ;               Otherwise, a Return2 will be made to the calling test.
;2295 ;
;2296 ;     b11   -   This code identifies the cases in which the test has
;2297 ;               been executed at least once (i.e., at least one of the
;2298 ;               controllers on the MS780-E unit under test could be
;2299 ;               configured properly).  When this code is detected the
;2300 ;               subroutine clears ID Reg 39 and makes a call to
;2301 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2302 ;               tested then the subroutine executes a Return1.
;2303 ;               Otherwise the subroutine is re-entered.
;2304 ;
;2305 ;
;2306 ; Calling Constraint: =00
;2307 ; ------------------
;2308 ;
;2309 ;
;2310 ; Inputs:
;2311 ; ------
;2312 ;
;2313 ; ID Register 39 must be cleared by NEWTST
;2314 ;
;2315 ;
;2316 ; Outputs:
;2317 ; -------
;2318 ;
;2319 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2320 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2321 ;         to be tested
            ;2322 ;
            ;2323 ; Side Effects:
            ;2324 ; ------------
            ;2325 ;
            ;2326 ; Contents of the following regiters will be destroyed:
            ;2327 ;
            ;2328 ; D, Q
            ;2329 ;
            ;2330 ;--
            ;2331 ;*************************************************************
            ;2332 =0
            ;2333 START.TEST:
U 105C, 0000,003D,0180,0800,0000,124E  ;2334  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 105D, 0000,003C,0180,0800,0000,1064  ;2335  NOP    ; ...tests that have more than one
            ;2336 =0      ; ...subtest.)
U 1064, 0000,003D,0180,0800,0000,122C  ;2337  ERLOOP    ; Set up the error loop for the
            ;2338       ; ...eventuality that the MS780-E/H
            ;2339       ; ...currently under test has no
            ;2340       ; ...Controllers
            ;2341 RE.ENTRY:    ; Re entry point for this routine
U 1065, 0000,003C,E5F0,2C00,0000,1240  ;2342  Q_ID[39]   ; Get the code
U 1240, 0C00,003C,0180,0800,0000,1241  ;2343  D_Q    ; Put it in the D Register
U 1241, 0000,193C,0180,0B00,0000,100C  ;2344  DI-0?    ; Branch on the code
U 100C, 0000,003C,0180,0B00,0000,1008  ;2345 =1100 J/STRT.CASE00   ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1010  ;2346  J/STRT.CASE01   ; Code is 01
U 100E, 0000,003C,0180,0800,0000,1246  ;2347  J/STRT.CASE10   ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1017  ;2348  J/STRT.CASE11   ; Code is 11
            ;2349 ;
            ;2350 ;
            ;2351 ; At this point the code in ID[39] was 00
            ;2352 ;
            ;2353 ;
            ;2354 =00
            ;2355 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1024  ;2356  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2357  J/STRT.LOW.MIS,   ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,1242  ;2358   D_K[.1]   ; Set up the code for re entry to this
            ;2359       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2360  D_K[.2]    ; Set up the code for a next call to
            ;2361       ; ...START.TEST indicating that the
            ;2362       ; ...Lower Controller was configured
            ;2363       ; ...( 10 )
            ;2364  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2365   RETURN2   ; Let the test know that the Lower
            ;2366       ; ...has been configured
            ;2367 STRT.LOW.MIS:
            ;2368  ID[39]_D,    ; Save code in ID[39] signifing that
U 1242, 0000,003C,E580,3C00,0000,1065  ;2369   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2370       ; ...and re enter this subroutine
            ;2371
            ;2372 ;
            ;2373 ;
            ;2374 ; At this point the code is 01
            ;2375 ;
```

```
            ;2376 ;
            ;2377 =00
            ;2378 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1028  ;2379  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2380  J/STRT.UPR.MIS,    ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,1243  ;2381   D_0    ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2382  D_K[.3]    ; Upper controller was configured
            ;2383       ; ...thus the code must be changed
            ;2384       ; ...accordingly ( 11 )
            ;2385  ID[39]_D,    ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2386   RETURN2    ; Let the test know that the Upper
            ;2387       ; ...Controller has been configured
            ;2388 STRT.UPR.MIS:
U 1243, 0000,003C,E580,3C00,0000,1244  ;2389  ID[39]_D    ; Save the Code ( 00 )
U 1244, 0018,0038,0D80,09E8,0000,1245  ;2390  RC[0D]_K[.3]    ; Set up a code for the Fail Chain
U 1245, 0000,003D,0980,0800,0084,704C  ;2391  ERROR1[.2]    ; Flag the error.
            ;2392 ;
            ;2393 ;
            ;2394 ; At this point the code is 10
            ;2395 ;
            ;2396 ;
            ;2397 STRT.CASE10:
U 1246, 0818,0038,0D80,0800,0000,1014  ;2398  D_K[.3]    ; Set the code to 11
            ;2399
            ;2400
            ;2401 =00 ID[39]_D,    ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1028  ;2402   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,1065  ;2403  J/RE.ENTRY    ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2404  RETURN2    ; Upper Controller configured
            ;2405       ; ...let the test know it
            ;2406 ;
            ;2407 ;
            ;2408 ; At this point the code is 11
            ;2409 ;
            ;2410 ;
            ;2411 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1020  ;2412  D_0    ; Clear the code
            ;2413 =00 ID[39]_D,    ; Save the code
U 1020, 0000,003D,E580,3C00,0000,12A4  ;2414   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2415       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,1065  ;2416  J/RE.ENTRY    ; Found another MS780-E/H.   Go and
            ;2417       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2418  RETURN1    ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1024  ;2419  NOP
```

```
;2420  .PAGE
;2421  .TOC "   SELECT LOWER CONTROLLER"
;2422 ;*************************************************************
;2423 ;**
;2424 ;
;2425 ; Functional Description:
;2426 ; ----------------------
;2427 ;
;2428 ; This subroutine can be called to select the lower
;2429 ; Controller on a MS780-E/H.
;2430 ; If the Lower controller is misconfigured then a
;2431 ; RETURN1 will be made.  Otherwise a RETURN2
;2432 ;
;2433 ; Calling Constraint: =00
;2434 ; ------------------
;2435 ;
;2436 ;
;2437 ; Inputs:
;2438 ; ------
;2439 ;
;2440 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2441 ;
;2442 ; Outputs:
;2443 ; -------
;2444 ;
;2445 ; RC[0D] will have the address of CNFG Register C
;2446 ;        ( 2000x008 )
;2447 ;
;2448 ; Side Effects:
;2449 ; ------------
;2450 ;
;2451 ; Contents of the following registers will lost:
;2452 ;
;2453 ; D
;2454 ;
;2455 ; The Lower controller on the MS780-E/H will be configured
;2456 ; when the subroutine is exitted with a RETURN2
;2457 ;--
;2458 ;*************************************************************
;2459 =00
;2460 SELECT.LOWER:
;2461  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,1236  ;2462   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2463  RETURN1    ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2464  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2465  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2466   RETURN2   ; Let caller know that the controller
;2467       ; ...was configured properly
```

```
;2468  .PAGE
;2469  .TOC "   SELECT UPPER CONTROLLER"
;2470 ;*************************************************************
;2471 ;**
;2472 ;
;2473 ; Functional Description:
;2474 ; ----------------------
;2475 ;
;2476 ; This subroutine can be called to select the upper
;2477 ; Controller on a MS780-E/H.
;2478 ; If the Upper controller is misconfigured then a
;2479 ; RETURN1 will be made.  Otherwise a RETURN2
;2480 ;
;2481 ; Calling Constraint: =00
;2482 ; ------------------
;2483 ;
;2484 ;
;2485 ; Inputs:
;2486 ; ------
;2487 ;
;2488 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2489 ;
;2490 ; Outputs:
;2491 ; -------
;2492 ;
;2493 ; RC[0D] will have the address of CNFG Register D
;2494 ;       ( 2000x010 )
;2495 ;
;2496 ; Side Effects:
;2497 ; ------------
;2498 ;
;2499 ; Contents of the following registers will lost:
;2500 ;
;2501 ; D
;2502 ;
;2503 ; The Upper controller on the MS780-E/H will be configured
;2504 ; when the subroutine is exitted with a RETURN2
;2505 ;--
;2506 ;*************************************************************
;2507 =00
;2508 SELECT.UPPER:
;2509  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,1236  ;2510   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2511  RETURN1     ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2512  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2513  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2514   RETURN2    ; Let caller know that the controller
;2515       ; ...was configured properly
```

```
            ;2516  .PAGE
            ;2517  .TOC "   Set Diagnostic Mode Routine"
            ;2518 ;++
            ;2519 ; FUNCTIONAL DESCRIPTION:
            ;2520 ;
            ;2521 ; This routine is used to set the specified diagnostic mode
            ;2522 ; in Register B. Other bits in the register remain unchanged.
            ;2523 ;
            ;2524 ; CALLING CONSTRAINT:
            ;2525 ;
            ;2526 ; =0
            ;2527 ;
            ;2528 ; INPUTS:
            ;2529 ;
            ;2530 ; d - Contains the mode to set in bits<1:0>
            ;2531 ; rc[0e] - Contains base address of controller
            ;2532 ;
            ;2533 ; SIDE EFFECTS:
            ;2534 ;
            ;2535 ; d,va,sc are destroyed
            ;2536 ;--
            ;2537 SET_DIAG_MODE:
U 1247, 0010,0038,D9F8,0970,0284,7248  ;2538  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 1248, 0D00,003C,0180,0803,0000,1249  ;2539  va_va+4,d_dal.sc ; Shift specified mode into position
U 1249, 0000,003C,01E0,0800,0000,124A  ;2540  q_d    ; Save the diagnostic mode bits
U 124A, 0000,003C,0180,C800,0000,124B  ;2541  d[long]_cache.p  ; Read the contents of the CNFG register B
U 124B, 081D,0030,0180,0800,0000,124C  ;2542  d_d.or.q  ; Set the diagnostic mode bits
U 124C, 0000,003E,0180,A800,0000,0001  ;2543  cache.p_d[long],return1 ; Set the specified mode and return
            ;2544
```

```
            ;2545  .PAGE
            ;2546  .TOC "   SBI Unjam Routine"
            ;2547 ;++
            ;2548 ; FUNCTIONAL DESCRIPTION:
            ;2549 ;
            ;2550 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2551 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2552 ; and finally HOLD for the last 16 cycles.
            ;2553 ;
            ;2554 ; CALLING CONSTRAINT:
            ;2555 ;
            ;2556 ; =0
            ;2557 ;
            ;2558 ; SIDE EFFECTS:
            ;2559 ;
            ;2560 ; D Reg destroyed
            ;2561 ;--
            ;2562   ;
            ;2563   ; assert hold for 16 cycles
            ;2564   ;
            ;2565 SBI.UNJAM:
            ;2566  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 124D, 0818,0038,6580,8000,0010,106C  ;2567      clk.ubcc  ; set micro cc's for next cycle
            ;2568 =0
            ;2569 SBI.UNJAM.1:
            ;2570  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2571       clk.ubcc,z?, ; ...
U 106C, 0819,0100,0580,8000,0010,106C  ;2572      j/sbi.unjam.1 ; test for completion
            ;2573   ;
            ;2574   ; assert hold and unjam for 16 cycles
            ;2575   ;
            ;2576  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 106D, 0818,0038,6580,8800,0010,10B8  ;2577      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2578 =0
            ;2579 SBI.UNJAM.2:
            ;2580  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2581       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10B8, 0819,0100,0580,8800,0010,10B8  ;2582      z?,j/sbi.unjam.2 ; ...
            ;2583   ;
            ;2584   ; assert hold for 16 cycles
            ;2585   ;
            ;2586  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10B9, 0818,0038,6580,8000,0010,10BA  ;2587      clk.ubcc  ; and set cc's for next cycle
            ;2588 =0
            ;2589 SBI.UNJAM.3:
            ;2590  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2591       clk.ubcc,z?, ; ...
U 10BA, 0819,0100,0580,8000,0010,10BA  ;2592      j/sbi.unjam.3 ; ...
U 10BB, 0000,003E,0180,0800,0000,0001  ;2593  return1   ; exit
```

```
            ;2594   .PAGE
            ;2595   .TOC "   RESET SUBTEST NUMBER"
            ;2596 ;**
            ;2597 ; FUNCTIONAL DESCRIPTION:
            ;2598 ;
            ;2599 ; Since some  of the tests will have to be restarted when two
            ;2600 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2601 ; the subtest counter to zero ( only for thoes tests that have
            ;2602 ; more than one subtest). This subroutine may also be necessary
            ;2603 ; when a test is being loop on.
            ;2604 ;
            ;2605 ; CALLING CONSTRAINT:
            ;2606 ;
            ;2607 ; =0
            ;2608 ; SIDE EFFECTS:
            ;2609 ;
            ;2610 ; D is destroyed
            ;2611 ;
            ;2612 ;--
            ;2613 RESET.SUBTST:
            ;2614   RC[0F]_0,    ; Reset subtest number
U 124E, 0003,003E,0180,09F8,0000,0001   ;2615   RETURN1    ; ...and return
```

```
            ;2616  .PAGE
            ;2617  .TOC "   Initialize the SBI"
            ;2618 ;++
            ;2619 ; FUNCTIONAL DESCRIPTION:
            ;2620 ;
            ;2621 ; This routine performs the following functions:
            ;2622 ;
            ;2623 ;  Executes an SBI Unjam
            ;2624 ;  Clears the SBI Fault Latch
            ;2625 ;  Clears the SBI Error Register
            ;2626 ;
            ;2627 ; CALLING CONSTRAINT:
            ;2628 ;
            ;2629 ; =0
            ;2630 ;
            ;2631 ; SIDE EFFECTS:
            ;2632 ;
            ;2633 ; D & Q Register destroyed
            ;2634 ;--
            ;2635 =0
            ;2636 SBI.INIT:
U 10BC, 0000.003D,0180.0800.0000.124D  ;2637  call[sbi.unjam]  ; Unjam the SBI
U 10BD, 0818.0038,C180.0800.0000.124F  ;2638  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 124F, 0801.0028,6580.3C00.0000.1250  ;2639  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1250, 0F00.003C,6D80.3C00.0000.1251  ;2640  id[fault]_d,d_0
U 1251, 0000.003C,6D80.3C00.0000.1252  ;2641  id[fault]_d
U 1252, 0000.003E,6580.3C00.0000.0001  ;2642  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
                ;2643  .PAGE
                ;2644  .TOC "   Disable Cache Routine"
                ;2645 ;++
                ;2646 ; FUNCTIONAL DESCRIPTION:
                ;2647 ;
                ;2648 ; This routine disables cache hits by setting bits <16:15>
                ;2649 ; in the maintenance register.
                ;2650 ;
                ;2651 ; CALLING SEQUENCE:
                ;2652 ;
                ;2653 ; =0
                ;2654 ;
                ;2655 ; SIDE EFFECTS:
                ;2656 ;
                ;2657 ; D & Q Registers destroyed
                ;2658 ;--
                ;2659 DISABLE.CACHE:
U 1253, 0818.0038.4580.0800.0000.1254  ;2660  d_k[.8000]  ; ... and seed constant
U 1254, 0500.003C.0180.0800.0000.1255  ;2661  d_d.left  ; Generate final constant
U 1255, 0819.0030.4580.0800.0000.1256  ;2662  d_d.or.k[.8000]  ; ... = 00018000
U 1256, 0000.003E.7580.3C00.0000.0001  ;2663  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2664  .PAGE
            ;2665  .TOC "   Enable Cache"
            ;2666 ;++
            ;2667 ; FUNCTIONAL DESCRIPTION:
            ;2668 ;
            ;2669 ; This routine enables cache group 0 by clearing the force
            ;2670 ; miss bit in the maintenance register.
            ;2671 ;
            ;2672 ; CALLING CONSTRAINT:
            ;2673 ;
            ;2674 ; =0
            ;2675 ;
            ;2676 ; SIDE EFFECTS:
            ;2677 ;
            ;2678 ; D, Q AND SC Registers destroyed
            ;2679 ;--
            ;2680 ENABLE.CACHE:
U 1257, 0000.003C.75F0.2C00.0000.1258  ;2681  q_id[maint]  ; Get current maintenance register
U 1258, 0000.003C.6580.0800.0084.7259  ;2682  sc_k[.10]  ; Setup to clear bit 16
U 1259, 0801.2034.0180.0800.0000.125A  ;2683  d_q.and.mask  ; Clear bit 16
U 125A, 0000.003E.7580.3C00.0000.0001  ;2684  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2685  .PAGE
            ;2686  .TOC "   Wait Loop Routine"
            ;2687 ;++
            ;2688 ; FUNCTIONAL DESCRIPTION:
            ;2689 ;
            ;2690 ; As the name implies,  this subroutine is used to set up a wait
            ;2691 ; loop for a specified  number of cycles.  This may be necessary
            ;2692 ; when the micro-program has to wait for another event to finish.
            ;2693 ; When the subroutine is entered Register D should be holding the
            ;2694 ; desired numbered of cycles.
            ;2695 ;
            ;2696 ; CALLING CONSTRAINT:
            ;2697 ;
            ;2698 ; =0
            ;2699 ;
            ;2700 ; INPUTS:
            ;2701 ;
            ;2702 ; d - Contains number of cycles to wait
            ;2703 ; alu.z condition code must not be set
            ;2704 ;
            ;2705 ; SIDE EFFECTS:
            ;2706 ;
            ;2707 ; d is destroyed
            ;2708 ;--
            ;2709 WAIT.LOOP:
U 125B, 0018,0038,6180,0800,0010,10BE  ;2710  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2711      ; ...is not set
            ;2712 =0
            ;2713 W.LOOP:
            ;2714  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10BE, 0819,0100,0580,0800,0010,10BE  ;2715       j/W.LOOP
U 10BF, 0000,003E,0180,0800,0000,0001  ;2716  return1    ; exit
            ;2717
```

```
            ;2718  .PAGE
            ;2719  .TOC "   Check for Interrupt Routine"
            ;2720 ;**
            ;2721 ; FUNCTIONAL DESCRIPTION:
            ;2722 ;
            ;2723 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2724 ; If an interrupt is active, the following registers are setup:
            ;2725 ;  RC[8] - Gets the VECTOR register
            ;2726 ;  RC[7] - Gets the SBI ERROR register
            ;2727 ;  RC[6] - Gets the FAULT register
            ;2728 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2729 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2730 ;
            ;2731 ; CALLING CONSTRAINT:
            ;2732 ;
            ;2733 ; =00
            ;2734 ;
            ;2735 ; SIDE EFFECTS:
            ;2736 ;
            ;2737 ; D & Q are destroyed
            ;2738 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2739 ;--
            ;2740 CHECK_INTERRUPT:
            ;2741    ;
            ;2742    ; First, enable the fault and RDS/CRD interrupts
            ;2743    ;
U 125C, 0818,0038,4580,0800,0000,125D  ;2744  d_k[.8000]  ; Get data to set interrupt enable
            ;2745  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 125D, 0500,003C,6580,3C00,0000,125E  ;2746      d_d.left
U 125E, 0100,003C,0180,0800,0000,125F  ;2747  d_d.left2  ; D = 40000
U 125F, 0000,003C,6D80,3C00,0000,1260  ;2748  id[fault]_d  ; Set fault interrupt enable
            ;2749  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1260, 0F03,003C,0180,09A8,4000,1261  ;2750      rc[5]_0  ; and clear interrupt occurred flag
U 1261, 0000,003C,3D80,3C00,0000,1262  ;2751  id[psl]_d  ; Clear the PSL
U 1262, 0000,003C,6580,3C00,0000,1263  ;2752  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1263, 0000,003C,6D80,3C00,0000,1264  ;2753  id[fault]_d  ; Clear FAULT Interrupt Enable
U 1264, 0000,0E3C,0180,0800,0000,1004  ;2754  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2755 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2756  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2757  return1   ; No interrupt
            ;2758 =111
            ;2759 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,1265  ;2760  q_id[vector]  ; Get ID Vector Register
U 1265, 0001,203C,65F0,2DC0,0000,1266  ;2761  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 1266, 0001,203C,6DF0,2DB8,0000,1267  ;2762  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 1267, 0001,203C,0180,09B0,0000,1268  ;2763  rc[6]_q   ; Save falut reg
U 1268, 0018,003A,0580,09A8,0000,0002  ;2764  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2765  .PAGE
            ;2766  .TOC "   CHECK UTRAP"
            ;2767 ;++
            ;2768 ;FUNCTIONAL DESCRIPTION:
            ;2769 ;
            ;2770 ; This subroutine is used to check wether a Utrap occured.
            ;2771 ; It takes the contents of the Save Utrap flags register
            ;2772 ; R[0E], and compares them to the contrnts of the Utrap
            ;2773 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2774 ; according to the results of the test  (i.e., contents of
            ;2775 ; these registers are equal or not).
            ;2776 ; CALLING CONSTRAINT:
            ;2777 ;
            ;2778 ; =00
            ;2779 ;
            ;2780 ; INPUTS:
            ;2781 ;
            ;2782 ; R[0E]- Saved Utrap Mask
            ;2783 ; R[0F]- Expected Utrap Mask
            ;2784 ;
            ;2785 ;--
            ;2786 CHECK_UTRAP:
U 1269, 0800,003C,0180,0A70,0000,126A  ;2787  D_R[0E]    ; Reg D is loaded with the trap mask
U 126A, 0001,003C,0180,09C0,0000,126B  ;2788  RC[08]_D   ; Save expected Utrap flag for error logout
U 126B, 0800,003C,0180,0A78,0000,126C  ;2789  D_R[0F]    ; Get recieved Utrap flag to D reg
U 126C, 0001,003C,0180,09B8,0000,126D  ;2790  RC[07]_D   ; Save in RC[07]
            ;2791  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 126D, 000D,0020,0180,0A70,0010,126E  ;2792          CLK.UBCC
U 126E, 0003,013C,0180,0AF8,0000,10C0  ;2793  Z?,R[0F]_0  ; Branch if no traps
U 10C0, 0000,003E,0180,0800,0000,0002  ;2794 =0 RETURN2   ; Utrap occured
U 10C1, 0000,003E,0180,0800,0000,0001  ;2795  RETURN1    ; No Utrap  return
            ;2796 ;
```

```
            ;2797  .PAGE
            ;2798  .TOC "   Set Trap Mask"
            ;2799 ;++
            ;2800 ; FUNCTIONAL DESCRIPTION:
            ;2801 ;
            ;2802 ; This routine is used to set a bit in the Micro Trap Mask
            ;2803 ; R[0F].
            ;2804 ;
            ;2805 ; CALLING CONSTRAINT:
            ;2806 ;
            ;2807 ; =0
            ;2808 ;
            ;2809 ; INPUTS:
            ;2810 ;
            ;2811 ; SC - Contains bit number to set.
            ;2812 ;
            ;2813 ; OUTPUTS:
            ;2814 ;
            ;2815 ; rc[0E] - Contains the current trap mask
            ;2816 ;
            ;2817 ; SIDE EFFECTS:
            ;2818 ;
            ;2819 ; d regsiter is destroyed
            ;2820 ;--
            ;2821 SET_TRAP_MASK:
U 126F, 0800,003C,0180,0A78,0000,1270  ;2822  d_r[0f]
U 1270, 0801,001C,0180,0AF8,0000,1271  ;2823  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1271, 0001,003E,0180,0AF0,0000,0001  ;2824  r[0E]_d,return1    ; Save mask and return
```

```
;2825  .PAGE
;2826  .TOC "   FIND MS780-E MEMORY"
;2827 ;**
;2828 ; FUNCTIONAL DESCRIPTION:
;2829 ;
;2830 ;          The diagnostic is designed to handle up to 4 (four)
;2831 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
;2832 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
;2833 ;     3D will hold the I/O base address of the MS780-E subsystems found
;2834 ;     on the SBI, and ID Register 3E will hold the Total number of
;2835 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
;2836 ;     first MS780-E found will have its starting address set to 0
;2837 ;     (zero).
;2838 ;     All the other MS780-E/H's found will have their starting address
;2839 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
;2840 ;     Reg 3E to decide if there are any more MS780-E and will take
;2841 ;     apropriate action.  Register RC[0E] is used as a pointer to the
;2842 ;     current MS780-E unit under test.
;2843 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
;2844 ;     set to maximum.
;2845 ;
;2846 ; CALLING CONSTRAINT:
;2847 ;
;2848 ; =00
;2849 ;
;2850 ; OUTPUTS:
;2851 ;
;2852 ; rc[0e] - Contains address of Configuration Register
;2853 ; r[0]  - Contains TR level
;2854 ;
;2855 ; SIDE EFFECTS:
;2856 ;
;2857 ; D register is destroyed.
;2858 ;--
;2859 =0
;2860 FIND.MS780E:
U 10C2, 0000,003D,0180,0800,0000,10BC  ;2861  call[sbi.init]  ; Make sure SBI is enabled
U 10C3, 0003,003C,0180,0988,0000,1272  ;2862  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1272, 0818,0038,1980,0800,0000,1273  ;2863  d_k[zero]  ; Clear MS780-E/H counter
U 1273, 0000,003C,F980,3C00,0000,1274  ;2864  id[3e]_d  ; ..
U 1274, 0018,0038,0580,0A80,0000,1275  ;2865  r[0]_k[.1]  ; Init TR counter
U 1275, 0F03,003C,0180,09F0,0000,10C4  ;2866  d_0,rc[0E]_0  ; Clear "Save" location for
;2867      ; ...CNFG A Reg
;2868 =0
;2869 FIND.MEM.LOOP:
U 10C4, 0800,003D,0180,0A00,0000,129B  ;2870  d_r[0],call[generate.io]; Generate address of TR level
U 10C5, 0001,003C,0180,09F0,0200,10C6  ;2871  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10C6, 0000,003D,8980,0800,0084,726F  ;2872 =0 set.trap.mask[.d] ; Enable timeout micro traps
;2873  d[long]_cache.p, ; Read the specified tr level
U 10C7, 0000,003C,0180,C970,0000,1276  ;2874   ic_rc[0e]  ; ...
U 1276, 0000,003C,0180,0A78,0010,1277  ;2875  alu_r[0f],clk.ubcc ; Check for no response
U 1277, 0001,013C,0180,0880,0082,10C8  ;2876  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10C8, 0000,003C,0180,0800,0000,1278  ;2877 =0 j/check.adpt.code ; Check for a memory
U 10C9, 0000,003C,0180,0800,0000,1297  ;2878  j/find.mem.lp1  ; Try next tr
;2879 CHECK.ADPT.CODE:
```

```
U 1278, 0000,003C,1D80,0800,1404,7279  ;2880  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1279, 0000,163C,0180,0800,0000,1018  ;2881  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,727A  ;2882 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,727B  ;2883  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,1297  ;2884  j/find.mem.lp1  ; Not a memory
U 101B, 0000,003C,0180,0800,0000,1297  ;2885  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,7280  ;2886  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,1297  ;2887  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,1284  ;2888  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,1284  ;2889  rc[0e]_lc,j/found.ms780e ; MS780-E
           ;2890   ;
           ;2891   ; Found an MS780-A or -C. Set the starting address
           ;2892   ; to maximum.
           ;2893   ;
U 127A, 0818,0038,5D80,0800,0000,127C  ;2894 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 127B, 0818,0038,0580,0800,0000,127C  ;2895 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
           ;2896 MS780.COMMON:
U 127C, 0819,0030,7180,0800,0000,127D  ;2897  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 127D, 0000,003C,01F8,0803,0080,D27E  ;2898  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 127E, 0D00,003C,0180,0800,0000,127F  ;2899  d_dal.sc  ; Put data in bits<29:14>
           ;2900  cache.p_d[long], ; Set the starting address to maximum
U 127F, 0000,003C,0180,A800,0000,1297  ;2901       j/find.mem.lp1 ; and continue TR scan
           ;2902   ;
           ;2903   ; Found an MA780. Set the starting address to maximum
           ;2904   ;
           ;2905 MA780.MAX:
U 1280, 0818,0038,39F8,0803,0000,1281  ;2906  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1281, 0D00,003C,0180,0803,0000,1282  ;2907  d_dal.sc,va_va+4 ; Put in proper position
U 1282, 0000,003C,0180,0803,0000,1283  ;2908  va_va+4   ; Point to Port Invalidate Ctrl Register
           ;2909  cache.p_d[long], ; Set starting address to maximum
U 1283, 0000,003C,0180,A800,0000,1297  ;2910       j/find.mem.lp1
           ;2911 ;
           ;2912 ; Found an MS780E
           ;2913 ;
           ;2914 FOUND.MS780E:
U 1284, 0000,003C,F9F0,2C00,0000,1285  ;2915  q_id[3e]  ; Set up to see how many MS780-E's
           ;2916  alu_q.xor.k[.3], ; Are there Maximum units?
U 1285, 0019,2020,0D80,0800,0010,1286  ;2917   clk.ubcc ; Check condition codes
U 1286, 0000,013C,0180,0800,0000,10CA  ;2918  z?   ; Are we at maximum units for system
U 10CA, 0000,003C,0180,0800,0000,1287  ;2919 =0 J/ms780e.tst  ; No go find how many units ther are
U 10CB, 0818,0038,C180,0800,0000,10CC  ;2920  d_k[.ffff]  ; Yes set address to maximum
U 10CC, 0000,003D,0180,0800,0000,129F  ;2921 =0 call[set.start.addr] ; .....
U 10CD, 0000,003C,0180,0800,0000,1297  ;2922  j/find.mem.lp1  ; Go check to see if we are done
           ;2923 ;
           ;2924 MS780E.TST:
           ;2925  alu_rc[01],  ; Check "Found MS780E Flag"
U 1287, 0010,0038,0180,0908,0010,1288  ;2926   clk.ubcc ; Check condition codes
U 1288, 0810,0138,0180,0970,0000,10CE  ;2927  z?,d_rc[0e]  ; Is this unit the first MS780E ?
           ;2928 =0 j/not.first.ms780e, ; No
U 10CE, 0818,0038,C180,0800,0000,10D2  ;2929   d_k[.ffff] ; Set starting address
U 10CF, 0001,003C,0180,0988,0000,1289  ;2930  rc[01]_d  ; Yes put base address in rc[01]
U 1289, 0818,0038,7980,0800,0000,128A  ;2931  d_k[.30]  ; Set up SC to point to first unit
U 128A, 0019,0014,F580,0800,0082,128B  ;2932  alu_d+k[.a],sc_alu ; ...
U 128B, 0810,0038,0180,0908,0000,128C  ;2933  d_rc[01]  ; Put base address of unit in D
U 128C, 0000,003C,0180,3400,0000,128D  ;2934  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 128D, 0F00,003C,0180,0800,0000,10D0  ;2935  d_0   ; Prepare to set starting address to Zero
U 10D0, 0000,003D,0180,0800,0000,129F  ;2936 =0 call[set.start.addr] ; Go do it
          ;2937  j/find.mem.lp1,  ; Check to see if we are done
U 10D1, 0003,003C,0180,09E8,0000,1297  ;2938   rc[0d]_0 ; ....
          ;2939 =0
          ;2940 NOT.FIRST.MS780E:
U 10D2, 000D,003D,0180,0800,0000,129F  ;2941  call[set.start.addr] ; Go set starting address to maximum
U 10D3, 0000,003C,F9F0,2C00,0000,128E  ;2942  q_id[3e]  ; Get the number of MS780-Es found
U 128E, 0819,2014,0580,0800,0000,128F  ;2943  d_q+k[.1]  ; Increment this counter
U 128F, 0000,003C,F980,3C00,0000,1290  ;2944  id[3e]_d  ; Save the counter
U 1290, 0018,0038,11C0,0800,0000,1291  ;2945  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2946     ; ...were the I/O base addresses will be stored
U 1291, 081D,2000,7980,0800,0000,1292  ;2947  d_q-d,k[.30]  ; ... adjust pointer
U 1292, 0819,0014,7980,0800,0000,1293  ;2948  d_d+k[.30]  ; Point the SC to the ID register
U 1293, 0000,003C,F580,0800,0000,1294  ;2949  k[.a]   ; ...to receive I/O base address
U 1294, 0019,0014,F580,0800,0082,1295  ;2950  alu_d+k[.a],sc_alu ; ...
U 1295, 0810,0038,0180,0970,0000,1296  ;2951  d_rc[0e]  ; Get base address to d
U 1296, 0000,003C,0180,3400,0000,1297  ;2952  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2953    ;
          ;2954    ; Try next tr
          ;2955    ;
          ;2956 FIND.MEM.LP1:
U 1297, 0818,0014,0580,0A80,0000,1298  ;2957  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2958  alu_d.and.k[.f],
U 1298, 0019,0034,6180,0800,0010,1299  ;2959      clk.ubcc  ; Check if all done
U 1299, 0000,013C,0180,0800,0000,10D4  ;2960  z?   ; Branch if yes
U 10D4, 0000,003C,0180,0800,0000,10C4  ;2961 =0 j/find.mem.loop  ; Try next TR
U 10D5, 0810,0038,0180,0908,0010,129A  ;2962  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 129A, 0000,013C,0180,0800,0000,10D6  ;2963  z?   ; Check condition codes
U 10D6, 0001,003E,0180,09F0,0000,0002  ;2964 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10D7, 0000,003E,0180,0800,0000,0001  ;2965  return1   ; No MS780E found
          ;2966
```

```
            ;2967  .PAGE
            ;2968  .TOC "   Generate IO Address"
            ;2969 ;++
            ;2970 ; FUNCTIONAL DESCRIPTION:
            ;2971 ;
            ;2972 ; This routine is used to generate an SBI Configuration Register
            ;2973 ; address from a TR level.
            ;2974 ;
            ;2975 ; CALLING CONSTRAINT:
            ;2976 ;
            ;2977 ; =0
            ;2978 ;
            ;2979 ; INPUTS:
            ;2980 ;
            ;2981 ; D - Contains TR level
            ;2982 ;
            ;2983 ; OUTPUTS:
            ;2984 ;
            ;2985 ; D - Contains SBI IO address
            ;2986 ;--
            ;2987 GENERATE.IO:
U 129B, 0000,003C,89F8,0800,0084,729C  ;2988  sc_k[.d],q_0   ; Setup to shift TR level 13 bits
U 129C, 0D00,003C,8D80,0800,0084,729D  ;2989  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 129D, 0000,003C,0980,0800,0084,B29E  ;2990  sc_sc-k[.2]   ; insert bit 29
U 129E, 0801,001E,0180,0800,0000,0001  ;2991  d_d.ornot.mask,return1 ; and return
```

```
            ;2992  .PAGE
            ;2993  .TOC "   Set Starting Address"
            ;2994 ;++
            ;2995 ; FUNCTIONAL DESCRIPTION:
            ;2996 ;
            ;2997 ; This routine is used to set the starting address of the
            ;2998 ; memory to the specified value.
            ;2999 ;
            ;3000 ; CALLING CONSTRAINT:
            ;3001 ;
            ;3002 ; =0
            ;3003 ;
            ;3004 ; INPUTS:
            ;3005 ;
            ;3006 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3007 ;     (B Register Image divided by 2**16)
            ;3008 ; rc[0e] - Contains Address of Register A
            ;3009 ;
            ;3010 ; OUTPUTS:
            ;3011 ;
            ;3012 ; d - Contains aligned starting address (B Register Image)
            ;3013 ;
            ;3014 ; SIDE EFFECTS:
            ;3015 ;
            ;3016 ; d,q,va, and sc are destroyed
            ;3017 ;--
            ;3018 SET.START.ADDR:
U 129F, 0010,0038,6580,0970,0284,72A0  ;3019  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 12A0, 0000,003C,01F8,0803,0000,12A1  ;3020  va_va+4,q_0  ; Set address to Register B
            ;3021  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 12A1, 0D00,003C,0980,0800,0084,B2A2  ;3022      sc_sc-k[.2]  ; Setup to insert bit 14
U 12A2, 0801,001C,0180,0800,0000,12A3  ;3023  d_d.ornot.mask  ; Insert Write Enable bit
U 12A3, 0000,003E,0180,A800,0000,0001  ;3024  cache.p_d[long],return1 ; Set the starting address and return
```

```
            ;3025  .PAGE
            ;3026  .TOC "   ENABLE NEXT MS780-E"
            ;3027 ;**
            ;3028 ; FUNCTIONAL DESCRIPTION:
            ;3029 ;
            ;3030 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3031 ;    They will be tested  separately, according to the TR level at which
            ;3032 ;    they are located,  starting with the one  at the lowest level first.
            ;3033 ;          When  a test has finished  with the first unit, it will have to call
            ;3034 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3035 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3036 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3037 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3038 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3039 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3040 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3041 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3042 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3043 ;
            ;3044 ; CALLING CONSTRAINT:
            ;3045 ;
            ;3046 ;  =00
            ;3047 ;
            ;3048 ;--
            ;3049 ENABLE.NEXT.MS780E:
U 12A4, 0000,003C,F9F0,2C00,0000,12A5  ;3050  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3051  ALU_Q,    ; Check to see if it is zero
U 12A5, 0001,203C,0180,0800,0010,12A6  ;3052      CLK.UBCC  ; Check Condition Codes
U 12A6, 0000,013C,0180,0800,0000,10D8  ;3053  Z?    ; ...for zero
U 10D8, 0000,003C,0180,0800,0000,12A7  ;3054 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10D9, 0000,003E,0180,0800,0000,0002  ;3055  RETURN2    ; Zero No more units to test
            ;3056 ENABLE.NEXT:
U 12A7, 0818,0038,C180,0800,0000,10DA  ;3057  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10DA, 0000,003D,0180,0800,0000,129F  ;3058 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10DB, 0818,0038,7980,0800,0000,12A8  ;3059  D_K[.30]  ; Prepare to point to the ID register holding
            ;3060      ; ...the I/O Base address of the next MS780-E
U 12A8, 0819,0014,1180,0800,0000,12A9  ;3061  D_D+K[.4]  ; ...
U 12A9, 0000,003C,F580,0800,0000,12AB  ;3062  K[.A]    ; ...
U 12AB, 0819,0014,F580,0800,0000,12AC  ;3063  D_D+K[.A]  ; ...
U 12AC, 0000,003C,F9F0,2C00,0000,12AD  ;3064  Q_ID[3E]  ; Get MS780-E Counter
            ;3065  D_D-Q,    ; D now holds the pointer to the ID register
U 12AD, 081D,0000,0180,0800,0082,12AE  ;3066    SC_ALU  ; ...
U 12AE, 0000,003C,01F0,2400,0000,12AF  ;3067  Q_ID(SC)  ; Q gets adrress of next MS780E
U 12AF, 0001,203C,0180,09F0,0000,12B0  ;3068  RC[0E]_Q  ; Save it also in RC[0E]
U 12B0, 0F03,003C,0180,09E8,0000,10DC  ;3069  RC[0D]_0,D_0  ; Set starting address to zero
U 10DC, 0000,003D,0180,0800,0000,129F  ;3070 =0 CALL[SET.START.ADDR] ; ....
U 10DD, 0F00,003C,F9F0,2C00,0000,12B1  ;3071  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 12B1, 081D,2008,0180,0800,0000,12B2  ;3072  D_Q-D-1    ; Subtract 1 from total
U 12B2, 0000,003C,F980,3C00,0000,10DE  ;3073  ID[3E]_D  ; Adjust the pointer
U 10DE, 0000,003D,0180,0800,0000,124E  ;3074 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10DF, 0000,003E,0180,0800,0000,0001  ;3075  RETURN1    ; Tell caller another unit was found
            ;3076
```

```
            ;3077  .PAGE
            ;3078  .TOC "   Set ECC Bits"
            ;3079 ;++
            ;3080 ; FUNCTIONAL DESCRIPTION:
            ;3081 ;
            ;3082 ; This routine is used to set the specified ECC bits
            ;3083 ; in Register B. Other bits in the register remain unchanged.
            ;3084 ;
            ;3085 ; CALLING CONSTRAINT:
            ;3086 ;
            ;3087 ; =0
            ;3088 ;
            ;3089 ; INPUTS:
            ;3090 ;
            ;3091 ; d - Contains the bits to set in <6:0>
            ;3092 ; rc[0] - Contains base address of controller
            ;3093 ;
            ;3094 ; SIDE EFFECTS:
            ;3095 ;
            ;3096 ; d,va,sc are destroyed
            ;3097 ;--
            ;3098 SET_ECC:
U 12B3, 0010.0038.0180.0970.0200.12B4  ;3099  va_rc[0E]  ; Set address of Register B
U 12B4, 0000.003C.0180.0803.0000.12B5  ;3100  va_va+4    ; ...in VA
U 12B5, 0000.003C.01E0.C800.0000.12B6  ;3101  q_d.d[long]_cache.p ; Read current contents of register
U 12B6, 0819.0024.5980.0800.0000.12B7  ;3102  d_d.andnot.k[.7f] ; Clear current ECC bits
U 12B7, 081D.0030.0180.0800.0000.12B8  ;3103  d_d.or.q  ; Insert new ecc bits
U 12B8, 0000.003E.0180.A800.0000.0001  ;3104  cache.p_d[long],return1 ; Set the specified bits and return
            ;3105
            ;3106
```

```
;3107  .PAGE
;3108  .TOC "    MS780-E/H CNFG REG CLEANUP"
;3109 ;*************************************************************
;3110 ;**
;3111 ;
;3112 ; Functional Description:
;3113 ; ----------------------
;3114 ;
;3115 ;      This subroutine is used to clear all error conditions
;3116 ; in the MS780-E/H Configuration/Status/Error Registers.
;3117 ; Bits beeing cleared are:
;3118 ;
;3119 ; CNFG Regsiter B <11:00>
;3120 ;
;3121 ; CNFG Register C and D <31:28> and <10:06>
;3122 ;   ---
;3123 ;
;3124 ;
;3125 ; Calling Constraint: =0
;3126 ; ------------------
;3127 ;
;3128 ;
;3129 ; Inputs:
;3130 ; ------
;3131 ;
;3132 ; RC[0E] MUST contain the I/O base address of
;3133 ; the MS780-E/H currently under test
;3134 ;
;3135 ; Outputs:
;3136 ; -------
;3137 ;
;3138 ; NO specific outputs.
;3139 ; Above mentioned bits will be cleared.
;3140 ;
;3141 ; Side Effects:
;3142 ; ------------
;3143 ;
;3144 ; The contents of the following registers will be lost
;3145 ; Q, D, SC, VA
;3146 ;
;3147 ;
;3148 ;--
;3149 ;*************************************************************
;3150 MS780E.CLEANUP:
U 12B9, 0010,0038,0180,0970,0200,12BA  ;3151  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
        ;3152  D_0,      ; Get data to clear Read/Write bits
U 12BA, 0F00,003C,0180,0803,0000,12BB  ;3153   VA_VA+4    ; Point to CNFG Reg B
U 12BB, 0000,003C,0180,A800,0000,12BC  ;3154  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 12BC, 0818,0038,7980,0800,0000,12BD  ;3155  D_K[.30]    ; Prepare to form x30000780 to clear
        ;3156        ; ...CNFG Reg's C and D
U 12BD, 0B00,003C,0180,0800,0000,12BE  ;3157  D_D.SWAP    ; D = x30000000
        ;3158  Q_D,      ; Save in the Q Reg
U 12BE, 0818,0038,CDE0,0800,0000,12BF  ;3159   D_K[.F0]   ; D = xF0
U 12BF, 0000,003C,0D80,0800,0084,72C0  ;3160  SC_K[.3]    ; Must shift D Reg left 3 bits to get
        ;3161       ; ...x780
```

```
U 12C0, 0D00.003C.0180.0800.0000.12C1  ;3162  D_DAL.SC   ; D = x780
        ;3163  D_D.OR.Q,   ; D = x30000780
U 12C1, 081D.0030.0180.0803.0000.12C2  ;3164   VA_VA+4   ; Point to CNFG Reg C
U 12C2, 0000.003C.0180.A800.0000.12C3  ;3165  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 12C3, 0000.003C.0180.0800.0000.12C4  ;3166  NOP
U 12C4, 0000.003C.0180.A800.0000.12C5  ;3167  CACHE.P_D[LONG]   ; Do second write to make sure that
        ;3168       ; ...uSequencer Parity error bit
        ;3169       ; ...is clear
U 12C5, 0000.003C.0180.0803.0000.12C6  ;3170  VA_VA+4    ; Point to CNFG Reg D
U 12C6, 0000.003C.0180.A800.0000.12C7  ;3171  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 12C7, 0000.003C.0180.0800.0000.12C8  ;3172  NOP
        ;3173  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
        ;3174       ; ...Parity error bit
U 12C8, 0000.003E.0180.A800.0000.0001  ;3175   RETURN1   ; Return to caller
        ;3176
```

```
            ;3177  .PAGE
            ;3178  .TOC "   CHECK SBI.ERR"
            ;3179 ;++
            ;3180 ; FUNCTIONAL DESCRIPTION:
            ;3181 ;
            ;3182 ; This subroutine can be used to see whether the SBI.ERR Register
            ;3183 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;3184 ; routine will have to pass to this subroutine in the SC register
            ;3185 ; the number of the error bit to checked.
            ;3186 ;
            ;3187 ; CALLING CONSTRAINT:
            ;3188 ;
            ;3189 ; =00
            ;3190 ;
            ;3191 ; INPUTS:
            ;3192 ;
            ;3193 ; SC with the number of the error bit to be checked
            ;3194 ;
            ;3195 ; OUTPUTS:
            ;3196 ;
            ;3197 ; RC[08]-Contains the SBI.ERROR Register
            ;3198 ;--
            ;3199 CHECK_SBI_ERR:
U 12C9, 0000,003C,65F0,2C00,0000,12CA  ;3200  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 12CA, 0001,203C,0180,09C0,0000,12CB  ;3201  RC[08]_Q  ; Save for error logout
            ;3202  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 12CB, 0001,2024,0180,0800,0010,12CC  ;3203   CLK.UBCC ; ...
U 12CC, 0000,013C,0180,0800,0000,10E0  ;3204  Z?   ; Check condition codes
U 10E0, 0000,003E,0180,0800,0000,0002  ;3205 =0 RETURN2   ; Bit is set
U 10E1, 0000,003E,0180,0800,0000,0001  ;3206  RETURN1   ; Bit is not set
            ;3207
```

```
            ;3208  .PAGE
            ;3209  .TOC "   Check Syndrome For Even Number of One's"
            ;3210 ;++
            ;3211 ; FUNCTIONAL DESCRIPTION:
            ;3212 ;
            ;3213 ; This routine is  used to check that the error syndrome
            ;3214 ; contains an even number of one's. This type of syndrome
            ;3215 ; occurs when there are exactly 2 bits in error.
            ;3216 ;
            ;3217 ; CALLING CONSTRAINT:
            ;3218 ;
            ;3219 ; =00
            ;3220 ;
            ;3221 ; INPUTS:
            ;3222 ;
            ;3223 ; D Reg - Contains the 7 bit syndrome zero extended
            ;3224 ;
            ;3225 ; OUTPUTS:
            ;3226 ;
            ;3227 ; RETURN1 - If number of one's is even.
            ;3228 ; RETURN2 - Otherwise
            ;3229 ;
            ;3230 ; SIDE EFFECTS:
            ;3231 ;
            ;3232 ; D, Q, and STATE are destroyed
            ;3233 ;--
            ;3234 SYNDROME.EVEN:
U 12CD, 0018,0038,5DC0,0800,0000,12CE  ;3235  q_k[.7]    ; Set a loop count
U 12CE, 0003,003C,0180,0800,1408,72CF  ;3236  state_0(A)  ; Set a flag
            ;3237 SYN.EVE.LOOP1:
U 12CF, 0000,193C,0180,0800,0000,102E  ;3238  d0?    ; Bit 0 a one?
            ;3239 =1110 d_d.right,  ; No
U 102E, 0600,003C,0180,0800,0000,12D0  ;3240   j/SYN.EVE.LOOP2
U 102F, 0600,003C,0180,0800,1400,D2D0  ;3241  d_d.right,state_state+1 ; Yes
            ;3242 SYN.EVE.LOOP2:
U 12D0, 0019,2000,05C0,0800,0010,12D1  ;3243  q_q-k[.1],clk.ubcc ; Tested all 7 bits?
U 12D1, 0000,013C,0180,0800,0000,10E2  ;3244  z?    ; Branch if yes
U 10E2, 0000,003C,0180,0800,0000,12CF  ;3245 =0 j/SYN.EVE.LOOP1   ; Check next bit
U 10E3, 0000,173C,0180,0800,0000,103E  ;3246  state0?    ; Even Number of one's?
U 103E, 0000,003E,0180,0800,0000,0001  ;3247 =1110 return1    ; Yes
U 103F, 0000,003E,0180,0800,0000,0002  ;3248  return2    ; No
            ;3249
            ;3250
```

```
;3251  .PAGE "TEST 1BD CHECK BITS GENERATION"
;3252 ;*************************************************************
;3253 ;+++
;3254 ;
;3255 ; Functional Description:
;3256 ; ----------------------
;3257 ;
;3258 ; This test makes use of Controller Diagnostic Mode 1 to verify the
;3259 ; correct functioning of the ECC generation circuitry.  In this mode,
;3260 ; memory Write/Read cycles are excuted normally except that on Writes
;3261 ; the check bits generated are also written to CNFG C (or D) register
;3262 ; bits <06:00>.  After writing the array, these bits are read back and
;3263 ; tested for correctness.  It is expected that the check bits read will
;3264 ; show a pattern of one floated in a field of zeroes.
;3265 ;
;3266 ; The data patterns will be stored in cache and they are as follows:
;3267 ;
;3268 ; Expected Check Bits Data Pattern
;3269 ; ------------------ ------------
;3270 ;
;3271 ;  01  00012224
;3272 ;  02  000DC36C
;3273 ;  04  000DC0CF
;3274 ;  08  000DC3EB
;3275 ;  10  0002FA69
;3276 ;  20  000FCD0C
;3277 ;  40  0000092A
;3278 ;
;3279  ;% 8D
;3280  ;$ 2224,0001,C36C,000D,C0CF,000D,C3EB,000D,FA69,0002,CD0C,000F,092A,0000
;3281 ;
;3282 ;
;3283 ; Logic Description:
;3284 ; ------------------
;3285 ;  N/A
;3286 ;
;3287 ; Error Logout:
;3288 ; ------------
;3289 ;
;3290 ; See individual error reports.
;3291 ;
;3292 ; Sync Point Description:
;3293 ; ----------------------
;3294 ;  N/A
;3295 ;
;3296 ;=============================================================
;3297 ;
;3298 ; Register Map:
;3299 ; ------------
;3300 ;
;3301 ;     RA,RB     Description:
;3302 ;     -----     -----------
;3303 ;
;3304 ;       0       Address of Data (longword) Patterns in Cache
;3305 ;
```

```
            ;3306 ;       1       Pattern Counter
            ;3307 ;
            ;3308 ;      RC       Description:
            ;3309 ;      --       -----------
            ;3310 ;
            ;3311 ; E I/O Base Address of MS780-E Currently Under Test
            ;3312 ;
            ;3313 ; D Configuration Register C or D (C for lower controller and
            ;3314 ;             D for upper controller)
            ;3315 ;
            ;3316 ; C Expected Check Bit Pattern
            ;3317 ;
            ;3318 ; B Received Check Bit Pattern
            ;3319 ;
            ;3320 ; A Memory Address Under Test
            ;3321 ;
            ;3322 ; 9 Data Pattern used to Generate Check Bit Pattern
            ;3323 ;
            ;3324 ;--
            ;3325 ;*************************************************************
            ;3326 =0
U 10E4, 0000,003D,0180,0800,0000,107F  ;3327 T.32: NEWTST   ; Signify start of new test
U 10E5, 0000,003C,0180,0800,0000,1030  ;3328  NOP
U 1030, 0000,003D,0180,0800,0000,10C2  ;3329 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1031, 0000,003C,0180,0800,0000,10F4  ;3330  J/T.32.EXIT  ; No MS780-E on the SBI, exit this test
U 1032, 0000,003C,0180,0800,0000,1034  ;3331  NOP
            ;3332 =
            ;3333 =00
            ;3334  ;
            ;3335  ; This point is entered after testing the first controller.  The
            ;3336  ; next Controller will be set-up and it will be tested.  IF there
            ;3337  ; are no more controllers then the test is exited.
            ;3338  ;
            ;3339 RESTART.TEST.32:
U 1034, 0000,003D,0180,0800,0000,105C  ;3340  CALL[START.TEST] ; Configure the controllers
U 1035, 0000,003C,0180,0800,0000,10F4  ;3341  J/T.32.EXIT  ; Got configuration error or no more
            ;3342      ; controllers to test, exit this test
U 1036, 0000,003D,0180,0800,0000,10BC  ;3343  CALL[SBI.INIT]  ; Initialize the SBI
U 1037, 0018,0038,5D80,0A88,0000,12D2  ;3344  R[1]_K[.7]  ; Set up Pattern Counter for 7 patterns
U 12D2, 0018,0038,4180,0A80,0000,12D3  ;3345  R[0]_K[.80]  ; Load R[0] with address of patterns in cache
U 12D3, 0003,003C,0180,09D0,0000,12D4  ;3346  RC[0A]_0  ; Load RC[0A] with test location
U 12D4, 0018,0038,0580,09E0,0000,10E6  ;3347  RC[0C]_K[.1]  ; Load RC[0C] with expected check bit pattern
            ;3348  ;
            ;3349  ; This loop point is for running the 7 different patterns through
            ;3350  ; the memory test location while checking the check bits generated for
            ;3351  ; each one of them.
            ;3352  ;
            ;3353  ; The first thing that is done is to get the longword of data from
            ;3354  ; cache that will be written to memory, which, will cause the check
            ;3355  ; bit under test to be asserted.
            ;3356  ;
            ;3357 =0
            ;3358 T.32.NEXT.PATTERN:
U 10E6, 0000,003D,0180,0800,0000,1257  ;3359  CALL[ENABLE.CACHE] ; Enable the cache
U 10E7, 0000,003C,0180,0A00,0200,12D5  ;3360  VA_R[0]   ; Get address of pattern stored in cache
```

```
ZZ-ESKAR-V22  TEST 1BD CHECK BITS GENERATION                                                    SEQ 1292
ESKAR4B.MCR                    MICRO2  1P(00)    26-JUL-85  17:28:07                          Page   9

U 12D5. 0000.003C.0180.C800.0000.12D6  ;3361  D[LONG]_CACHE.P  ; Read the pattern and
U 12D6. 0001.003C.0180.09C8.0000.12D7  ;3362  RC[09]_D  ; Store it in RC[09] in case of error
U 12D7. 0800.003C.0180.0A00.0000.12D8  ;3363  D_R[0]   ; Move cache pointer to the D register
U 12D8. 0019.0014.1180.0A80.0000.10E8  ;3364  R[0]_D+K[.4]  ; Update the cache pointer for next pattern
U 10E8. 0000.003D.0180.0800.0000.1253  ;3365 =0 CALL[DISABLE.CACHE] ; Disable the cache
U 10E9. 0000.003C.0180.0800.0000.10EA  ;3366  NOP
          ;3367  ;
          ;3368  ; Now that we got the data pattern, set diagnostic mode to 1 for
          ;3369  ; checking the CHECK bits, then write the test pattern out to memory.
          ;3370  ;
U 10EA. 0818.0039.0580.0800.0000.1247  ;3371 =0 SET.DIAG.MODE[.1] ; Select diagnostic mode of 1
U 10EB. 0000.003C.0180.0800.0000.10EC  ;3372  NOP
U 10EC. 0000.003D.0180.0800.0000.122C  ;3373 =0 ERLOOP   ; Loop on error from here
U 10ED. 0810.0038.0180.0948.0000.12D9  ;3374  D_RC[09]  ; Move test pattern to the D reg for writing
U 12D9. 0010.0038.0180.0950.0200.12DA  ;3375  VA_RC[0A]  ; Load VA with the test address
U 12DA. 0000.003C.0180.A800.0000.12DB  ;3376  CACHE.P_D[LONG]  ; Write the test pattern
          ;3377  ;
          ;3378  ; Now after writing the test pattern to memory, read the contents of
          ;3379  ; Configuraton register C or D (dependent on whether the upper or lower
          ;3380  ; controller is being tested) and mask out the Check bits.
          ;3381  ;
U 12DB. 0010.0038.0180.0968.0200.12DC  ;3382  VA_RC[0D]  ; Load VA with address of Err Register C or D
          ;3383     ; C for lower controller and D for upper
U 12DC. 0000.003C.0180.C800.0000.12DD  ;3384  D[LONG]_CACHE.P  ; Read the register
U 12DD. 0019.0034.5980.09D8.0000.12DE  ;3385  RC[0B]_D.AND.K[.7F] ; Mask & store the check bits from Err register
U 12DE. 0810.0038.0180.0960.0000.12DF  ;3386  D_RC[0C]  ; Load the D register with expected check bit
          ;3387     ; pattern
U 12DF. 0011.0020.0180.0958.0010.12E0  ;3388  ALU_D.XOR.RC[0B],CLK.UBCC
          ;3389     ; Exclusive Or expected check bit pattern with
          ;3390     ; the received check bits.
U 12E0. 0000.013C.0180.0800.0000.10EE  ;3391  Z?   ; Are the Check Bits correct ?
U 10EE. 0000.003C.0180.0800.0000.10F0  ;3392 =0 J/T.33.CHK.BIT.ERROR ; No, Call an error
U 10EF. 0000.003C.0180.0800.0000.10F1  ;3393  J/T.33.CHK.BIT.OK ; Yes, go shift the expected pattern by 1 and
          ;3394     ; continue testing.
```

```
            ;3395  .PAGE
            ;3396 ;
            ;3397 ;////////////////////////////////////////////////////////////////////////////////
            ;3398 ;
            ;3399 ; Received check bits do not match the expected.
            ;3400 ; Call an error.
            ;3401 ;
            ;3402 =0
            ;3403 T.33.CHK.BIT.ERROR:
U 10F0, 0000,003D,D580,0800,0084,7054  ;3404  ERROR2[.6]
            ;3405 ;
            ;3406 ;////////////////////////////////////////////////////////////////////////////////
            ;3407 ;
            ;3408 ; ERROR LOGOUT:
            ;3409 ;
            ;3410 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3411 ;  Configuration Register C or D (C for lower controller and
            ;3412 ;              D for upper controller)
            ;3413 ;  Expected Check Bit Pattern
            ;3414 ;  Received Check Bit Pattern
            ;3415 ;  Memory Address Under Test
            ;3416 ;  Data Pattern used to Generate Check Bit Pattern
            ;3417 ;
            ;3418 ;////////////////////////////////////////////////////////////////////////////////
            ;3419 ;
            ;3420  ;
            ;3421  ; Come here after the check bits generated matched the check bits
            ;3422  ; expected.  Shift the expected check bit pattern and decrement the
            ;3423  ; pattern count.  If the pattern count in equal to 0 then restart
            ;3424  ; this test for the next controller, ELse continue with the next
            ;3425  ; pattern.
            ;3426  ;
            ;3427 T.33.CHK.BIT.OK:
U 10F1, 0810,0038,0180,0960,0000,12E1  ;3428  D_RC[0C]  ; Move expected check bit pattern to the D reg
            ;3429  SI/ASHL,  ; Specify a shift of zero's into low bit and
U 12E1, 0021,003C,0100,09E0,0000,12E2  ;3430  RC[0C]_D.LEFT  ; shift expected check bit pattern left on bit
U 12E2, 0800,003C,0180,0A08,0000,12E3  ;3431  D_R[1]   ; Move pattern count to the D register
            ;3432  ALU_D-K[.1],  ; Decrement the pattern count and
U 12E3, 0019,0000,0580,0A88,0010,12E4  ;3433  R[1]_ALU,CLK.UBCC ; store back in R[1]
U 12E4, 0000,013C,0180,0800,0000,10F2  ;3434  Z?   ; Is pattern count equal to zero ?
U 10F2, 0000,003C,0180,0800,0000,10E6  ;3435 =0 J/T.32.NEXT.PATTERN ; No, do the next pattern
U 10F3, 0000,003C,0180,0800,0000,1034  ;3436  J/RESTART.TEST.32 ; yes, restart this test for the next
            ;3437     ; controller.
            ;3438  ;
            ;3439  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;3440  ;
            ;3441 =0
            ;3442 T.32.EXIT:
U 10F4, 0000,003D,0180,0800,0000,12B9  ;3443  CALL[MS780E.CLEANUP] ; Clear memory CNFG Regs before exitting
U 10F5, 0000,003C,0180,0800,0000,10F6  ;3444  NOP
            ;3445
```

```
;3446  .PAGE "TEST 1BE SINGLE BIT ERROR DETECTION"
;3447 ;************************************************************
;3448 ;+++
;3449 ;
;3450 ; Functional Description:
;3451 ; ---------------------
;3452 ;
;3453 ; This test verifies the detection of CRD errors.  It is performed by
;3454 ; setting the Controller in Diagnostics Mode 2, and forcing a CRD error.
;3455 ; It is expected that: a CRD tag will be sent on the SBI, bit 9 in CNFG
;3456 ; Register C/D is set and the syndrome bits are latched in CNFG C
;3457 ; register.
;3458 ;
;3459 ; The test floats a one across a field of zeroes and a zero accross a
;3460 ; field of ones while forcing CRD errors.  The syndrome bits will be
;3461 ; stored in cache, an they are as follows:
;3462 ;
;3463  ;% 40
;3464  ;$ 4A4F,5452,5857,5D5B,2523,2926,2C2A,3431
;3465  ;$ 0B0E,1513,1916,1C1A,6462,6867,6D6B,7570
;3466 ;
;3467 ;   SUBTEST 1
;3468 ;
;3469 ; This subtest checks whether a CRD bit can be detected in any bit
;3470 ; position while the bit in error is a one.
;3471 ;
;3472 ;   SUBTEST 2
;3473 ;
;3474 ; This subtest is similar to the previous one except that in this case
;3475 ; the bit in error is a zero.
;3476 ;
;3477 ;
;3478 ; Logic Description:
;3479 ; -----------------
;3480 ;  N/A
;3481 ;
;3482 ; Error Logout:
;3483 ; ------------
;3484 ;
;3485 ; See individual error reports.
;3486 ;
;3487 ; Sync Point Description:
;3488 ; ----------------------
;3489 ;  N/A
;3490 ;
;3491 ;============================================================
;3492 ;
;3493 ; Register Map:
;3494 ; ------------
;3495 ;
;3496 ;     RA,RB     Description:
;3497 ;     -----     -----------
;3498 ;
;3499 ;       2 Pattern Counter
;3500 ;
```

```
            ;3501 ;       1       Shift Designator Flag  1 = Shift Left w/ones in
            ;3502 ;      0 = Shift Left w/zero's in
            ;3503 ;
            ;3504 ;       0 Pointer for Cache Address of Syndromes
            ;3505 ;
            ;3506 ;      RC       Description:
            ;3507 ;      --       -----------
            ;3508 ;
            ;3509 ;       E I/O Base Address of MS780-E Currently Under Test
            ;3510 ;
            ;3511 ;       D I/O Address of Configuration Register C/D ( Lower/Upper)
            ;3512 ;
            ;3513 ;       C Expected Syndrome Bits
            ;3514 ;
            ;3515 ;       B Received Syndrome Bits (Contents of CNFG C/D)
            ;3516 ;
            ;3517 ;       A Memory Test Address
            ;3518 ;
            ;3519 ;       9 Storage of Pattern Used to Force Error
            ;3520 ;
            ;3521 ;--
            ;3522 ;*****************************************************************
            ;3523 =0
U 10F6, 0000,003D,0180,0800,0000,107F  ;3524 T.33: NEWTST   ; Signify start of new test
U 10F7, 0000,003C,0180,0800,0000,1038  ;3525  NOP
U 1038, 0000,003D,0180,0800,0000,10C2  ;3526 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1039, 0000,003C,0180,0800,0000,1136  ;3527  J/T.33.EXIT  ; No MS780-E's to test, exit
U 103A, 0000,003C,0180,0800,0000,1040  ;3528  NOP
            ;3529 =
            ;3530  ;
            ;3531  ; This is the point in the test where control is transferred to when
            ;3532  ; the test has completed for each controller and it is necessary to
            ;3533  ; restart the test for the next controller.
            ;3534  ;
            ;3535 =00
            ;3536 RESTART.TEST.33:
U 1040, 0000,003D,0180,0800,0000,105C  ;3537  CALL[START.TEST] ; Configure Controllers
U 1041, 0000,003C,0180,0800,0000,1136  ;3538  J/T.33.EXIT  ; No more controllers to test, exit this test
U 1042, 0000,003C,0180,0800,0000,1043  ;3539  NOP   ; More controllers, continue with test
            ;3540 T.33.START.HERE:
U 1043, 0003,003C,0180,09D0,0000,12E5  ;3541  RC[0A]_0  ; Load RC[0A] with test address to reference
            ;3542 ;
            ;3543 ;=================================================================================
            ;3544 ;
            ;3545 ; Subtest 1
            ;3546 ;
U 12E5, 0018,0038,3180,0A80,0000,12E6  ;3547  R[0]_K[.40]  ; Load R[0] with cache address of syndromes
U 12E6, 0003,003C,0180,0A88,0000,12E7  ;3548  R[1]_0   ; Shift designation flag ( 1 = shift left
            ;3549     ; w/one's and 0 = shift left w/zero's)
U 12E7, 0018,0038,0580,09C8,0000,10F8  ;3550  RC[09]_K[.1]  ; Load pattern to force the error
U 10F8, 0000,003D,0180,0800,0000,10FC  ;3551 =0 CALL[TEST.33.TST] ; Execute the test
            ;3552 ;
            ;3553 ;=================================================================================
            ;3554 ;
            ;3555 ; Subtest 2
```

```
            ;3556 ;
U 10F9, 0018,0038,3180,0A80,0000,12E8  ;3557  R[0]_K[.40]  ; Load R[0] with cache address of syndromes
U 12E8, 0018,0038,0580,0A88,0000,12E9  ;3558  R[1]_K[.1]  ; Set shift designation flag
U 12E9, 0000,003C,1980,0800,0084,72EA  ;3559  SC_K[ZERO]  ; Put zero in SC for masking bit 0 = 0 and all
            ;3560      ; others = 1's
U 12EA, 0801,0038,0180,0800,0000,12EB  ;3561  D_D[B]MASK  ; Set all bits in mask except for bit <0> and
U 12EB, 0001,003C,0180,09C8,0000,10FA  ;3562  RC[09]_D  ; Store in RC[09]
U 10FA, 0000,003D,0180,0800,0000,10FC  ;3563 =0 CALL[TEST.33.TST] ; Execute the test
U 10FB, 0000,003C,0180,0800,0000,1040  ;3564  J/RESTART.TEST.33 ; Restart test for the next controller
            ;3565  ;
            ;3566  ; This is the start of the test code that is called as a subroutine.
            ;3567  ;
            ;3568 =0
            ;3569 TEST.33.TST:
U 10FC, 0000,003D,0180,0800,0000,1234  ;3570  SUBTEST  ; Update the Subtest counter
U 10FD, 0018,0038,7580,0A90,0000,10FE  ;3571  R[2]_K[.20]  ; Set up for 32(decimal) patterns
            ;3572  ;
            ;3573  ; This is the entry point for the 32 passes necessary to float the bit
            ;3574  ; across a 32 bit longword
            ;3575  ;
            ;3576 =0
            ;3577 T.33.LOOP.POINT:
U 10FE, 0000,003D,0180,0800,0000,10BC  ;3578  CALL[SBI.INIT]  ; Initialize the SBI
U 10FF, 0000,003C,0180,0800,0000,110A  ;3579  NOP
            ;3580  ;
            ;3581  ; Set-up to write the test pattern and then write it out to the memory
            ;3582  ; location that is under test.
            ;3583  ;
U 110A, 0000,003D,0180,0800,0000,12B9  ;3584 =0 CALL[MS780E.CLEANUP] ; Clear MS780-E/H CNFG Registers
U 110B, 0000,003C,0180,0800,0000,1110  ;3585  NOP
U 1110, 0818,0039,8580,0800,0000,12B3  ;3586 =0 SET.ECC[.C]  ; Set ECC substitute bits to xC for data of all
            ;3587      ; zero's
U 1111, 0010,0038,0180,0950,0200,12EC  ;3588  VA_RC[0A]  ; Load VA with test address to be accessed
U 12EC, 0810,0038,0180,0948,0000,12ED  ;3589  D_RC[09]  ; Move test pattern to the D register for write
U 12ED, 0000,003C,0180,A800,0000,1112  ;3590  CACHE.P_D[LONG]  ; Write the test pattern
U 1112, 0818,0039,0980,0800,0000,1247  ;3591 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode 2
U 1113, 0000,003C,0180,0800,0000,1114  ;3592  NOP
U 1114, 0000,003D,0180,0800,0000,122C  ;3593 =0 ERLOOP  ; Loop on error from here
U 1115, 0000,003C,0180,0800,0000,1116  ;3594  NOP
            ;3595  ;
            ;3596  ; Now that we wrote the pattern, read the pattern back and check to
            ;3597  ; see if there is a RDS micro trap generated from the read.  There
            ;3598  ; shouldn't be so if there is call an error.
            ;3599  ;
U 1116, 0000,003D,8580,0800,0084,726F  ;3600 =0 SET.TRAP.MASK[.C] ; Enable Read Data Substitute micro traps
U 1117, 0010,0038,0180,0950,0200,12EE  ;3601  VA_RC[0A]  ; Load VA with the test address
U 12EE, 0000,003C,0180,C800,0000,12EF  ;3602  D[LONG]_CACHE.P  ; Read the test location to force the error
U 12EF, 0010,0038,0180,0968,0200,1048  ;3603  VA_RC[0D]  ; Point to CNFG Register C/D
U 1048, 0000,003D,0180,0800,0000,1269  ;3604 =00 CHECK.UTRAP  ; Was there a micro trap ?
U 1049, 0000,003C,0180,0800,0000,1119  ;3605  J/T.33.NO.UTRAP  ; No, continue testing
U 104A, 0000,003C,0180,0800,0000,1118  ;3606  J/T.33.RDS.UTRAP ; Yes, shouldn't have got RDS utrap, call error
            ;3607 =
```

```
            ;3608  .PAGE
            ;3609 ;
            ;3610 ;////////////////////////////////////////////////////////////////////////////////
            ;3611 ;
            ;3612 ; RDS micro trap on read.
            ;3613 ; Call an error.
            ;3614 ;
            ;3615 =0
            ;3616 T.33.RDS.UTRAP:
U 1118. 0000,003D,0180,0800,0084,7054  ;3617  ERROR2[.8]
            ;3618 ;
            ;3619 ;////////////////////////////////////////////////////////////////////////////////
            ;3620 ;
            ;3621 ; ERROR LOGOUT:
            ;3622 ;
            ;3623 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3624 ;  I/O Address of Configuration Register C/D
            ;3625 ;  Unused
            ;3626 ;  Unused
            ;3627 ;  Memory Test Address
            ;3628 ;  Pattern Used to Force Error
            ;3629 ;  Expected Micro Trap Flags
            ;3630 ;  Received Micro Trap Flags
            ;3631 ;
            ;3632 ;////////////////////////////////////////////////////////////////////////////////
            ;3633 ;
            ;3634  ;
            ;3635  ; Didn't get a micro trap, now check bit 9 of the Configuration Register
            ;3636  ; C/D to make sure that the CRD error bit got set.  If it didn't call
            ;3637  ; an error.
            ;3638  ;
            ;3639 T.33.NO.UTRAP:
U 1119. 0000,003C,0180,C800,0000,12F0  ;3640  D[LONG]_CACHE.P  ; Read the contents of CNFG C/D
U 12F0. 0001,003C,0180,09C0,0000,12F1  ;3641  RC[08]_D  ; Save contents of CNFG C/D in case error
U 12F1. 0000,003C,D980,0800,0084,72F2  ;3642  SC_K[.9]  ; Get ready to mask bit 9
U 12F2. 0003,001C,0180,09E0,0000,12F3  ;3643  RC[0C]_NOT.MASK  ; Expected bit to be set in CNFG reg C/D
            ;3644  D_D.ANDNOT.MASK, ; Check to see if the bit is set
U 12F3. 0801,0024,0180,0800,0010,12F4  ;3645   CLK.UBCC
            ;3646      ;
U 12F4. 0001,013C,0180,09D8,0000,111A  ;3647  Z?,RC[0B]_D  ; Was bit 9 set ?
U 111A. 0000,003C,0180,0800,0000,111D  ;3648 =0 J/T.33.BIT9.SET  ; Yes, should have been, continue with test
U 111B. 0000,003C,0180,0800,0000,111C  ;3649  NOP   ; No, should have been, call an error
```

```
            ;3650  .PAGE
            ;3651 ;
            ;3652 ;////////////////////////////////////////////////////////////////////////////////
            ;3653 ;
            ;3654 ; Bit 9 ( CRD Flag ) in Configuration Register C/D was NOT set.
            ;3655 ; Call an error.
            ;3656 ;
            ;3657 =0
U 111C. 0000,003D,5D80,0800,0084,7054  ;3658  ERROR2[.7]
            ;3659 ;
            ;3660 ;////////////////////////////////////////////////////////////////////////////////
            ;3661 ;
            ;3662 ; ERROR LOGOUT:
            ;3663 ;
            ;3664 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3665 ;  I/O Address of Configuration Register C/D
            ;3666 ;  Expected bit to be set in CNFG Reg C/D
            ;3667 ;  Received status of the bit in CNFG Reg C/D
            ;3668 ;  Memory Test Address
            ;3669 ;  Pattern Used to Force Error
            ;3670 ;  Contents of Configuration Register C/D
            ;3671 ;
            ;3672 ;////////////////////////////////////////////////////////////////////////////////
            ;3673 ;
            ;3674  ;
            ;3675  ; The CNFG C/D register bit <9> was ok, now check to see that the cpu
            ;3676  ; logged the fact that there was a CRD in memory.  This is done by
            ;3677  ; the memory sending a CRD Tag back with the read data.
            ;3678  ;
            ;3679 T.33.BIT9.SET:
U 111D. 0818,0038,0180,0800,0000,111E  ;3680  D_K[.8]   ; Load D with the number of cycles to wait for
            ;3681     ; the SBI to finish the cycle
U 111E. 0000,003D,0180,0800,0000,125B  ;3682 =0 CALL[WAIT.LOOP]  ; Wait the specified (D) number of cycles.
U 111F. 0000,003C,0180,0800,0000,12F5  ;3683  NOP
U 12F5. 0000,003C,8980,0800,0084,7050  ;3684  SC_K[.D]  ; Load SC with xD
            ;3685 =00 SC_SC+1,  ; Increment the SC for bit position xE
U 1050. 0000,003D,0180,0800,0080,D2C9  ;3686  CALL[CHECK_SBI_ERR] ; and check to see if that bit is set or clear
            ;3687     ; in the ID - SBI.ERR register.
U 1051. 0000,003C,0180,0800,0000,1120  ;3688  J/T.33.BIT14.NOTSET ; Bit 14 (xE) was not set and should have been
            ;3689     ; call an error
U 1052. 0000,003C,0180,0800,0000,1122  ;3690  J/T.33.BIT14.SET ; Bit 14 (xE) was set continue testing.
            ;3691 =
```

```
            ;3692  .PAGE
            ;3693 ;
            ;3694 ;////////////////////////////////////////////////////////////////////////////////
            ;3695 ;
            ;3696 ; SBI ERROR register bit 14 (CRD) was not set and should have been.
            ;3697 ; Call an error.
            ;3698 ;
            ;3699 =0
            ;3700 T.33.BIT14.NOTSET:
U 1120, 0000.003D.5D80.0800.0084.7054  ;3701  ERROR2[.7]
U 1121, 0000.003C.0180.0800.0000.1122  ;3702  NOP
            ;3703 ;
            ;3704 ;////////////////////////////////////////////////////////////////////////////////
            ;3705 ;
            ;3706 ; ERROR LOGOUT:
            ;3707 ;
            ;3708 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3709 ;  I/O Address of Configuration Register C/D
            ;3710 ;  Unused
            ;3711 ;  Unused
            ;3712 ;  Memory Test Address
            ;3713 ;  Pattern Used to Force Error
            ;3714 ;  ID[sbi.err] Register
            ;3715 ;
            ;3716 ;////////////////////////////////////////////////////////////////////////////////
            ;3717 ;
            ;3718  ;
            ;3719  ; Ok, now lets check to see that the syndrome generated from the forced
            ;3720  ; CRD error was correct.  Read the CNFG C/D register and compare it to
            ;3721  ; the expected syndrome bits, if not equal then call an error.
            ;3722  ;
            ;3723 =0
            ;3724 T.33.BIT14.SET:
U 1122, 0000.003D.0180.0800.0000.1257  ;3725  CALL[ENABLE.CACHE] ; Turn on Cache
U 1123, 0000.003C.0180.0A00.0200.12F6  ;3726  VA_R[0]   ; Load VA with address of Syndromes in Cache
U 12F6, 0000.803C.0180.C800.0000.12F7  ;3727  D[BYTE]_CACHE.P  ; Read one byte (syndrome) from cache.
            ;3728  ALU_D.AND.K[.7F], ; Mask byte 0
U 12F7, 0019.0034.5980.09E0.0000.12F8  ;3729  RC[0C]_ALU  ; Save in RC[0C]
U 12F8, 0800.003C.0180.0A00.0000.12F9  ;3730  D_R[0]   ; Load D with the Cache pointer
U 12F9, 0019.0014.0580.0A80.0000.1124  ;3731  R[0]_D+K[.1]  ; Update the Cache pointer
U 1124, 0000.003D.0180.0800.0000.1253  ;3732 =0 CALL[DISABLE.CACHE] ; Disable the Cache
U 1125, 0000.003C.0180.0800.0000.1126  ;3733  NOP
U 1126, 0000.003D.0180.0800.0000.10BC  ;3734 =0 CALL[SBI.INIT]  ; Initialize the Sbi
U 1127, 0000.003C.0180.0800.0000.1128  ;3735  NOP
U 1128, 0818.0039.8580.0800.0000.12B3  ;3736 =0 SET.ECC[.C]  ; Set ECC substitute bits to xC for data of all
            ;3737     ; 0's.
U 1129, 0010.0038.0180.0950.0200.12FA  ;3738  VA_RC[0A]  ; Point VA to the test address
U 12FA, 0810.0038.0180.0948.0000.12FB  ;3739  D_RC[09]  ; Load D with the test pattern
U 12FB, 0000.003C.0180.A800.0000.112A  ;3740  CACHE.P_D[LONG]  ; Write the test pattern to the location
U 112A, 0818.0039.0980.0800.0000.1247  ;3741 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode 2
U 112B, 0000.003C.0180.0800.0000.112C  ;3742  NOP
U 112C, 0000.003D.0180.0800.0000.122C  ;3743 =0 ERLOOP   ; Loop on error from here
U 112D, 0010.0038.0180.0950.0200.12FC  ;3744  VA_RC[0A]  ; Load VA with the test address
U 12FC, 0000.003C.0180.C800.0000.12FD  ;3745  D[LONG]_CACHE.P  ; Read and force the CRD error
U 12FD, 0010.0038.0180.0968.0200.12FE  ;3746  VA_RC[0D]  ; Load VA with Configuration Register C/D
```

```
U 12FE, 0000,003C,0180,C800,0000,12FF  ;3747  D[LONG]_CACHE.P  ; Read the contents of CNFG C/D
U 12FF, 0001,003C,0180,09D8,0000,1300  ;3748  RC[0B]_D  ; Save read data in RC[0B]
U 1300, 0810,0038,0180,0958,0000,1301  ;3749  D_RC[0B]  ; Load D with save contents of CNFG C/D
        ;3750  ALU_D.AND.K[.7F], ; Mask out bits <6:0> from CNFG C/D
U 1301, 0019,0034,5980,09D8,0000,1302  ;3751  RC[0B]_ALU  ;
U 1302, 0810,0038,0180,0960,0000,1303  ;3752  D_RC[0C]  ; Load D with expected syndrome
        ;3753  ALU_D.XOR.RC[0B], ; Exclusive Or received syndrome bits with
U 1303, 0011,0020,0180,0958,0010,1304  ;3754   CLK.UBCC ; ... expected syndrome bits
U 1304, 0000,013C,0180,0800,0000,112E  ;3755  Z?   ; Were the syndromes equal ?
U 112E, 0000,003C,0180,0800,0000,1130  ;3756 =0 J/T.33.SYNDROME.NEQ ; No, Call an error
U 112F, 0000,003C,0180,0800,0000,1131  ;3757  J/T.33.SYNDROME.EQ ; Yes, continue with the test
```

```
            ;3758  .PAGE
            ;3759 ;
            ;3760 ;////////////////////////////////////////////////////////////////////////////////
            ;3761 ;
            ;3762 ; Received syndrome and expected syndrome did not match.  Call an error.
            ;3763 ;
            ;3764 =0
            ;3765 T.33.SYNDROME.NEQ:
U 1130, 0000,003D,D580,0800,0084,7054  ;3766  ERROR2[.6]
            ;3767 ;
            ;3768 ;////////////////////////////////////////////////////////////////////////////////
            ;3769 ;
            ;3770 ; ERROR LOGOUT:
            ;3771 ;
            ;3772 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3773 ;  I/O Address of Configuration Register C/D
            ;3774 ;  Expected Syndrome Bits
            ;3775 ;  Received Syndrome Bits
            ;3776 ;  Memory Test Address
            ;3777 ;  Pattern Used to Force Error
            ;3778 ;
            ;3779 ;////////////////////////////////////////////////////////////////////////////////
            ;3780 ;
            ;3781  ;
            ;3782  ; Expected and Received syndrome was correct, now check the shift
            ;3783  ; designation flag to see if shifting the pattern left with ones
            ;3784  ; shifted in or if shifting left with zero's in.
            ;3785  ;
            ;3786 T.33.SYNDROME.EQ:
U 1131, 0000,003C,0180,0A08,0010,1305  ;3787  ALU_R[1],CLK.UBCC ; Load ALU with shift designator bit
U 1305, 0000,013C,0180,0800,0000,1132  ;3788  Z?   ; Is this bit 0 or 1
U 1132, 0000,003C,0180,0800,0000,1307  ;3789 =0 J/T.33.LEFT.WONES ; Is a 1, shift left with ones in
            ;3790  ;
            ;379   ; Shift Left with Zero's in
            ;37.2  ;
U 1133, 0810,0038,0180,0948,0000,1306  ;3793  D_RC[09]  ; Is a 0, shift left with zero's in.  Load
            ;3794      ; pattern to be shifted into the D register
            ;3795  SI/MUL+,  ; Specify a shift in of Zero and
            ;3796  RC[09]_D.LEFT,  ; perform the shift left one place
U 1306, 0021,003C,0300,09C8,0000,1309  ;3797  J/T.33.DECREMENT.R2 ; Now go decrement the pattern counter
            ;3798  ;
            ;3799  ; Shift Left with One's in
            ;3800  ;
            ;3801 T.33.LEFT.WONES:
U 1307, 0810,0038,0180,0948,0000,1308  ;3802  D_RC[09]  ; Load pattern to shift into the D register
            ;3803  SI/MUL-,  ; Specify a shift in of One and
U 1308, 0021,003C,0380,09C8,0000,1309  ;3804  RC[09]_D.LEFT  ; perform the shift left one place, then fall
            ;3805      ; through to decrement R[2].
            ;3806  ;
            ;3807  ; Decrement the pass count and if equal to 0 after the decrement then
            ;3808  ; return to the caller.
            ;3809  ;
            ;3810 T.33.DECREMENT.R2:
U 1309, 0800,003C,0180,0A10,0000,130A  ;3811  D_R[2]   ; Move pattern counter to the D register
            ;3812  ALU_D-K[.1],  ; Decrement the pattern counter,
```

```
            ;3813  R[2]_ALU,   ; restore back in R[2], and then
U 130A, 0019,0000,0580,0A90,0010,130B  ;3814  CLK.UBCC  ; clock condition codes to check value
U 130B, 0000,013C,0180,0800,0000,1134  ;3815  Z?    ; Is counter now equal to Zero ?
U 1134, 0000,003C,0180,0800,0000,10FE  ;3816 =0 J/T.33.LOOP.POINT ; No, return to loop point for another pass
U 1135, 0000,003E,0180,0800,0000,0001  ;3817  RETURN1   ; Yes, return to the caller
            ;381B  ;
            ;3819  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;3820  ;
            ;3821 =0
            ;3822 T.33.EXIT:
U 1136, 0000,003D,0180,0800,0000,12B9  ;3823  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
U 1137, 0000,003C,0180,0800,0000,1138  ;3824  NOP
            ;3825
```

```
;3826  .PAGE "TEST 1BF SINGLE BIT ERROR CORRECTION PART 1"
;3827 ;************************************************************
;3828 ;***
;3829 ;
;3830 ; Functional Description:
;3831 ; ----------------------
;3832 ;
;3833 ; This test verifies that a single bit error in any bit position is
;3834 ; corrected properly.  The test makes use of Diagnostic Mode 2.  It
;3835 ; floats a one and a zero through location 0 of the array board while
;3836 ; forcing CRD error on Reads using bits <06:00> of CNFG B register,
;3837 ; using as substitute ECC bits those for data of zero (or all ones).
;3838 ; The data returned from memory and the data on the array is checked for
;3839 ; correctness.
;3840 ;
;3841 ;   SUBTEST 1
;3842 ;
;3843 ; In this subtest a one will be floated accross a field of zeroes.  The
;3844 ; expected data to be returned from the memory is of all 0's.
;3845 ;
;3846 ;   SUBTEST 2
;3847 ;
;3848 ; This subtest floats a zero accroxx a field of ones, and the expected
;3849 ; data to be returned from the memory is all 1's.
;3850 ;
;3851 ; Logic Description:
;3852 ; -----------------
;3853 ;  N/A
;3854 ;
;3855 ; Error Logout:
;3856 ; ------------
;3857 ;
;3858 ; See individual error reports.
;3859 ;
;3860 ; Sync Point Description:
;3861 ; ----------------------
;3862 ;  N/A
;3863 ;
;3864 ;============================================================
;3865 ;
;3866 ; Register Map:
;3867 ; ------------
;3868 ;
;3869 ;     RA,RB     Description:
;3870 ;     -----     -----------
;3871 ;
;3872 ;       0       Initial Memory Test Pattern
;3873 ;
;3874 ;       1       Expected Memory Data
;3875 ;
;3876 ;       2 Pattern Counter
;3877 ;
;3878 ;      RC       Description:
;3879 ;      --       -----------
;3880 ;
```

```
            ;3881 ;       E I/O Base Address of MS780-E Currently Under Test
            ;3882 ;
            ;3883 ; D I/O Address of Configuration Register C/D
            ;3884 ;
            ;3885 ;       C Expected Memory Data
            ;3886 ;
            ;3887 ;       B Received Memory Data
            ;3888 ;
            ;3889 ; A Memory Test Address
            ;3890 ;
            ;3891 ;       9 Memory Init Pattern
            ;3892 ;
            ;3893 ;--
            ;3894 ;*****************************************************************
            ;3895 =0
U 1138. 0000.003D.0180.0800.0000.107F  ;3896 T.34: NEWTST   ; Signify start of new test
U 1139. 0000.003C.0180.0800.0000.1060  ;3897  NOP
U 1060. 0000.003D.0180.0800.0000.10C2  ;3898 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the system
U 1061. 0000.003C.0180.0800.0000.115A  ;3899  J/T.34.EXIT  ; No MS780-E's found, exit this test
U 1062. 0000.003C.0180.0800.0000.1068  ;3900  NOP
            ;3901 =
            ;3902 =00
            ;3903  ;
            ;3904  ; This entry point is used to restart the test after completed testing
            ;3905  ; a controller and ready to configure/enable the next controller.
            ;3906  ;
            ;3907 RESTART.TEST.34:
U 1068. 0000.003D.0180.0800.0000.105C  ;3908  CALL[START.TEST] ; Configure the Controllers
U 1069. 0000.003C.0180.0800.0000.115A  ;3909  J/T.34.EXIT  ; No more controllers to configure
U 106A. 0003.003C.0180.09D0.0000.106B  ;3910  RC[0A]_0  ; Load RC[0A] with address on the array to
            ;3911     ; be accessed by this test
            ;3912 ;
            ;3913 ;=================================================================
            ;3914 ;
            ;3915 ; Subtest 1
            ;3916 ;
U 106B. 0018.0038.0580.0A80.0000.130C  ;3917  R[0]_K[.1]  ; Load R[0] with initial memory pattern
U 130C. 0003.003C.0180.0A88.0000.113A  ;3918  R[1]_0   ; Expected Returned memory data
U 113A. 0000.003D.0180.0800.0000.113E  ;3919 =0 CALL[TEST.34.TST] ; Execute the test
            ;3920 ;
            ;3921 ;=================================================================
            ;3922 ;
            ;3923 ; Subtest 2
            ;3924 ;
U 113B. 001B.001C.0580.0A80.0000.130D  ;3925  R[0]_NOT.K[.1]  ; Load initial memory pattern (FFFFFFFE)
U 130D. 001B.001C.1980.0A88.0000.113C  ;3926  R[1]_NOT.K[ZERO] ; Expected memory data
U 113C. 0000.003D.0180.0800.0000.113E  ;3927 =0 CALL[TEST.34.TST] ; Execute the test
U 113D. 0000.003C.0180.0800.0000.1068  ;3928  J/RESTART.TEST.34 ; Restart this test for next controller
            ;3929  ;
            ;3930  ; The following code is the actual test code but is called as a
            ;3931  ; subroutine to prevent repitition of code.
            ;3932  ;
            ;3933 =0
            ;3934 TEST.34.TST:
U 113E. 0000.003D.0180.0800.0000.1234  ;3935  SUBTEST   ; Update the subtest pointer
```

```
U 113F, 0800.003C.0180.0A00.0000.130E  ;3936  D_R[0]    ; Load D with pattern to initialize memory
U 130E, 0001.003C.0180.09C8.0000.130F  ;3937  RC[09]_D   ; then store it in RC[09]
U 130F, 0800.003C.0180.0A08.0000.1310  ;3938  D_R[1]    ; Load D with expected bit pattern from array
U 1310, 0001.003C.0180.09E0.0000.1311  ;3939  RC[0C]_D   ; then store it in RC[0C]
U 1311, 0018.0038.7580.0A90.0000.1312  ;3940  R[2]_K[.20]  ; Set-up a counter for 32 different bit
          ;3941      ; patterns
          ;3942  ;
          ;3943  ; This is the entry point for the 32 passes necessary to float a bit
          ;3944  ; through a 32 bit longword.
          ;3945  ;
          ;3946 T.34.LOOP.POINT:
U 1312, 0010.0038.0180.0950.0200.1313  ;3947  VA_RC[0A]  ; Load VA with memory test address
U 1313, 0810.0038.0180.0948.0000.1314  ;3948  D_RC[09]  ; Move init pattern to D for writing
U 1314, 0000.003C.0180.A800.0000.1140  ;3949  CACHE.P_D[LONG]  ; Write the init pattern
U 1140, 0000.003D.0180.0800.0000.10BC  ;3950 =0 CALL[SBI.INIT]  ; Initialize the Sbi
U 1141, 0000.003C.0180.0800.0000.1142  ;3951  NOP
U 1142, 0818.0039.0980.0800.0000.1247  ;3952 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode 2
U 1143, 0000.003C.0180.0800.0000.1144  ;3953  NOP
U 1144, 0818.0039.8580.0800.0000.12B3  ;3954 =0 SET.ECC[.C]  ; Set the ECC bits for data of all zeroes
U 1145, 0000.003C.0180.0800.0000.1146  ;3955  NOP
U 1146, 0000.003D.0180.0800.0000.122C  ;3956 =0 ERLOOP   ; Loop on error from here
U 1147, 0000.003C.0180.0800.0000.1148  ;3957  NOP
          ;3958  ;
          ;3959  ; Read the memory location to force the CRD error.
          ;3960  ;
U 1148, 0000.003D.8580.0800.0084.726F  ;3961 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 1149, 0010.0038.0180.0950.0200.1315  ;3962  VA_RC[0A]  ; Load VA with memory test address
U 1315, 0000.003C.0180.C800.0000.1316  ;3963  D[LONG]_CACHE.P  ; Read the location forcing the error
U 1316, 0001.003C.0180.09D8.0000.1070  ;3964  RC[0B]_D  ; Save read data in RC[0B]
          ;3965  ;
          ;3966  ; Check for any micro traps.  There shouldn't be so if there is then
          ;3967  ; call an error
          ;3968  ;
U 1070, 0000.003D.0180.0800.0000.1269  ;3969 =00 CHECK.UTRAP  ; Was the a RDS micro trap ?
U 1071, 0000.003C.0180.0800.0000.114B  ;3970  J/T.34.CHK.READ.1 ; No, continue with the test
U 1072, 0000.003C.0180.0800.0000.114A  ;3971  J/T.34.RDS.UTRAP ; Yes, call an error
          ;3972 =
```

```
            ;3973  .PAGE
            ;3974 ;
            ;3975 ;////////////////////////////////////////////////////////////////////////////////
            ;3976 ;
            ;3977 ; Got a RDS micro trap when reading the memory test location.
            ;3978 ; Call an error.
            ;3979 ;
            ;3980 =0
            ;3981 T.34.RDS.UTRAP:
U 114A, 0000,003D,0180,0800,0084,7054  ;3982  ERROR2[.8]
            ;3983 ;
            ;3984 ;////////////////////////////////////////////////////////////////////////////////
            ;3985 ;
            ;3986 ; ERROR LOGOUT:
            ;3987 ;
            ;3988 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3989 ;  I/O Address of Configuration Register C/D
            ;3990 ;  Expected Memory Data
            ;3991 ;  Received Memory Data
            ;3992 ;  Memory Test Address
            ;3993 ;  Memory Init Pattern
            ;3994 ;  Expected Micro Trap Flags
            ;3995 ;  Received Micro Trap Flags
            ;3996 ;
            ;3997 ;////////////////////////////////////////////////////////////////////////////////
            ;3998 ;
            ;3999  ;
            ;4000  ; Seeing how there was no micro traps now lets check the data we read
            ;4001  ; and see if it is correct.
            ;4002  ;
            ;4003 T.34.CHK.READ.1:
U 114B, 0810,0038,0180,0958,0000,1317  ;4004  D_RC[0B]  ; Move Received memory data to the D register
            ;4005  ALU_D.XOR.RC[0C], ; Exclusive Or the received data with the
U 1317, 0011,0020,0180,0960,0010,1318  ;4006  CLK.UBCC  ; expected data.
U 1318, 0000,013C,0180,0800,0000,114C  ;4007  Z?   ; Did the received and expected data match ?
U 114C, 0000,003C,0180,0800,0000,114E  ;4008 =0 J/T.34.BAD.DATA.1 ; No, call an error
U 114D, 0000,003C,0180,0800,0000,1150  ;4009  J/T.34.DO.SEC.READ ; Yes, continue with the test
```

```
            ;4010  .PAGE
            ;4011 ;
            ;4012 ;////////////////////////////////////////////////////////////////////////////////
            ;4013 ;
            ;4014 ; Received data did not match Expected data. (i.e., memory did not correct
            ;4015 ; the data it returned)
            ;4016 ; Call an error.
            ;4017 ;
            ;4018 =0
            ;4019 T.34.BAD.DATA.1:
U 114E, 0000.003D.D580.0800.0084.7054  ;4020  ERROR2[.6]
U 114F, 0000.003C.0180.0800.0000.1150  ;4021  NOP
            ;4022 ;
            ;4023 ;////////////////////////////////////////////////////////////////////////////////
            ;4024 ;
            ;4025 ; ERROR LOGOUT:
            ;4026 ;
            ;4027 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4028 ;  I/O Address of Configuration Register C/D
            ;4029 ;  Expected Memory Data
            ;4030 ;  Received Memory Data
            ;4031 ;  Memory Test Address
            ;4032 ;  Memory Init Pattern
            ;4033 ;
            ;4034 ;////////////////////////////////////////////////////////////////////////////////
            ;4035 ;
            ;4036  ;
            ;4037  ; Ok, now set diagnostic mode to ECC bypass and read the data from the
            ;4038  ; array location to make sure that it was stored correctly.
            ;4039  ;
            ;4040 =0
            ;4041 T.34.DO.SEC.READ:
U 1150, 0818.0039.0D80.0800.0000.1247  ;4042  SET.DIAG.MODE[.3] ; Select ECC bypass mode
U 1151, 0010.0038.0180.0950.0200.1319  ;4043  VA_RC[0A]  ; Load VA with memory address under test
U 1319, 0000.003C.0180.C800.0000.131A  ;4044  D[LONG]_CACHE.P  ; Read the memory location under test
U 131A, 0001.003C.0180.09D8.0000.131B  ;4045  RC[0B]_D  ; Save read data in RC[0B]
            ;4046  ALU_D.XOR.RC[0C], ; Exclusive Or the received data and the
U 131B, 0011.0020.0180.0960.0010.131C  ;4047  CLK.UBCC  ; expected data.
U 131C, 0000.013C.0180.0800.0000.1152  ;4048  Z?  ; Does received and expected data match ?
U 1152, 0000.003C.0180.0800.0000.1154  ;4049 =0 J/T.34.BAD.DATA.2 ; No, call an error
U 1153, 0000.003C.0180.0800.0000.1155  ;4050  J/T.34.CHK.R1  ; Yes, continue with the test
```

```
            ;4051   .PAGE
            ;4052 ;
            ;4053 ;////////////////////////////////////////////////////////////////////////////
            ;4054 ;
            ;4055 ; Received data and Expected data did not match.  (i.e., data was not
            ;4056 ; corrected on the array)
            ;4057 ; Call an error.
            ;4058 ;
            ;4059 =0
            ;4060 T.34.BAD.DATA.2:
U 1154, 0000,003D,D580,0800,0084,7054  ;4061  ERROR2[.6]
            ;4062 ;
            ;4063 ;////////////////////////////////////////////////////////////////////////////
            ;4064 ;
            ;4065 ; ERROR LOGOUT:
            ;4066 ;
            ;4067 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4068 ;  I/O Address of Configuration Register C/D
            ;4069 ;  Expected Memory Data
            ;4070 ;  Received Memory Data
            ;4071 ;  Memory Test Address
            ;4072 ;  Memory Init Pattern
            ;4073 ;
            ;4074 ;////////////////////////////////////////////////////////////////////////////
            ;4075 ;
            ;4076 T.34.CHK.R1:
U 1155, 0000,003C,0180,0A08,0010,131D  ;4077  ALU_R[1],CLK.UBCC ; Load ALU with shift designator bit
U 131D, 0000,013C,0180,0800,0000,1156  ;4078  Z?   ; Is this bit 0 or 1
U 1156, 0000,003C,0180,0800,0000,131F  ;4079 =0 J/T.34.LEFT.WONES ; Is a 1, shift left with ones in
            ;4080  ;
            ;4081  ; Shift Left with Zero's in
            ;4082  ;
U 1157, 0810,0038,0180,0948,0000,131E  ;4083  D_RC[09]  ; Is a 0, shift left with zero's in.  Load
            ;4084      ; pattern to be shifted into the D register
            ;4085  SI/MUL+,  ; Specify a shift in of Zero and
            ;4086  RC[09]_D.LEFT,  ; perform the shift left one place
U 131E, 0021,003C,0300,09C8,0000,1321  ;4087  J/T.34.DECREMENT.R2 ; Now go decrement the pattern counter
            ;4088  ;
            ;4089  ; Shift Left with One's in
            ;4090  ;
            ;4091 T.34.LEFT.WONES:
U 131F, 0810,0038,0180,0948,0000,1320  ;4092  D_RC[09]  ; Load pattern to shift into the D register
            ;4093  SI/MUL-,  ; Specify a shift in of One and
U 1320, 0021,003C,0380,09C8,0000,1321  ;4094  RC[09]_D.LEFT  ; perform the shift left one place, then fall
            ;4095      ; through to decrement R[2].
            ;4096  ;
            ;4097  ; Decrement the pass count and if equal to 0 after the decrement then
            ;4098  ; return to the caller.
            ;4099  ;
            ;4100 T.34.DECREMENT.R2:
U 1321, 0800,003C,0180,0A10,0000,1322  ;4101  D_R[2]   ; Move pattern counter to the D register
            ;4102  ALU_D-K[.1],  ; Decrement the pattern counter,
            ;4103  R[2]_ALU,  ; restore back in R[2], and then
U 1322, 0019,0000,0580,0A90,0010,1323  ;4104  CLK.UBCC  ; clock condition codes to check value
U 1323, 0000,013C,0180,0800,0000,1158  ;4105  Z?   ; Is counter now equal to Zero ?
```

```
U 1158, 0000,003C,0180,0800,0000,1312  ;4106 =0 J/T.34.LOOP.POINT ; No, return to loop point for another pass
U 1159, 0000,003E,0180,0800,0000,0001  ;4107  RETURN1   ; Yes, return to the caller
          ;4108  ;
          ;4109  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
          ;4110  ;
          ;4111 =0
          ;4112 T.34.EXIT:
U 115A, 0000,003D,0180,0800,0000,12B9  ;4113  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
U 115B, 0000,003C,0180,0800,0000,115C  ;4114  NOP
          ;4115
```

```
;4116  .PAGE "TEST 1C0 SINGLE BIT ERROR CORRECTION PART 2"
;4117 ;*************************************************************
;4118 ;+++
;4119 ;
;4120 ; Functional Description:
;4121 ; ---------------------
;4122 ;
;4123 ; For this test the array location 0 is initialized to 0 or to all ones
;4124 ; as the case may be.  Single bit errors are forced by using Diagnostic
;4125 ; mode 2 and by setting the substitute ECC bits in CNFG B register in
;4126 ; such a manner that when the data is read from the array a one will be
;4127 ; floated across a field of zeroes ( or its complement).  The ECC bit
;4128 ; pattern that will be used in this test will be stored in cache and they
;4129 ; are as follows:
;4130 ;
;4131  ;% C0
;4132  ;$ 4643,585E,545B,5157,292F,252A,2026,383D
;4133  ;$ 0702,191F,151A,1016,6B6E,646B,6167,797C
;4134 ;
;4135 ;   SUBTEST 1
;4136 ;
;4137 ; In this subtest a one will be floated across a field of zeroes.  The
;4138 ; expected data to be returned from the memory is of all 0's.
;4139 ;
;4140 ;   SUBTEST 2
;4141 ;
;4142 ; This subtest floats a zero across a field of one's, and teh expected
;4143 ; data to be returned from the memory is all 1's.
;4144 ;
;4145 ;
;4146 ; Logic Description:
;4147 ; -----------------
;4148 ;  N/A
;4149 ;
;4150 ; Error Logout:
;4151 ; ------------
;4152 ;
;4153 ; See individual error reports.
;4154 ;
;4155 ; Sync Point Description:
;4156 ; ----------------------
;4157 ;  N/A
;4158 ;
;4159 ;*************************************************************
;4160 ;
;4161 ; Register Map:
;4162 ; ------------
;4163 ;
;4164 ;     RA,RB     Description:
;4165 ;     -----     -----------
;4166 ;
;4167 ;       0       Memory Initialize Pattern
;4168 ;
;4169 ;       1       Expected Memory Return Data
;4170 ;
```

```
            ;4171 ;       2 Pattern Counter
            ;4172 ;
            ;4173 ;       3 Cache Address of Check Bit PatternsOM
            ;4174 ;
            ;4175 ;      RC       Description:
            ;4176 ;      --       -----------
            ;4177 ;
            ;4178 ;       E I/O Base Address of MS780-E Currently Under Test
            ;4179 ;
            ;4180 ;       D I/O Address of Configuration Register C/D
            ;4181 ;
            ;4182 ;       C Expected Memory Data
            ;4183 ;
            ;4184 ;       B Received Memory Data
            ;4185 ;
            ;4186 ; A Memory Test Address
            ;4187 ;
            ;4188 ;       9 Memory Initialize Pattern
            ;4189 ;
            ;4190 ;       8 Check Bit Pattern for SET.ECC
            ;4191 ;
            ;4192 ;--
            ;4193 ;*************************************************************
            ;4194 =0
U 115C, 0000,003D,0180,0800,0000,107F  ;4195 T.35: NEWTST   ; Signify start of new test
U 115D, 0000,003C,0180,0800,0000,1074  ;4196  NOP
U 1074, 0000,003D,0180,0800,0000,10C2  ;4197 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the system
U 1075, 0000,003C,0180,0800,0000,1182  ;4198  J/T.35.EXIT  ; No MS780-E's on the system to test
U 1076, 0000,003C,0180,0800,0000,1078  ;4199  NOP
            ;4200 =
            ;4201
            ;4202  ;
            ;4203  ; This is the entry point for when the test completes one controller
            ;4204  ; and is ready to go to the next (if there is a next).  Come here
            ;4205  ; and enable the next controller.  If no more controllers then exit
            ;4206  ; this test
            ;4207  ;
            ;4208 =00
            ;4209 RESTART.TEST.35:
U 1078, 0000,003D,0180,0800,0000,105C  ;4210  CALL[START.TEST] ; Configure controllers
U 1079, 0000,003C,0180,0800,0000,1182  ;4211  J/T.35.EXIT  ; No more controllers to configure
U 107A, 0003,003C,0180,09D0,0000,107B  ;4212  RC[0A]_0  ; Load RC[0A] with memory test address
            ;4213 ;
            ;4214 ;=====================================================================
            ;4215 ;
            ;4216 ; Subtest 1
            ;4217 ;
U 107B, 0003,003C,0180,0A80,0000,1324  ;4218  R[0]_0   ; Initial memory Pattern
U 1324, 0018,0038,0580,0A88,0000,1325  ;4219  R[1]_K[.1]  ; Expected returned memory data
U 1325, 0018,0038,D180,0A98,0000,115E  ;4220  R[3]_K[.C0]  ; Load R[3] with the cache address of the check
            ;4221      ; bit patterns
U 115E, 0000,003D,0180,0800,0000,1162  ;4222 =0 CALL[TEST.35.TST] ; Execute the test
            ;4223 ;
            ;4224 ;=====================================================================
            ;4225 ;
```

```
            ;4226 ; Subtest 2
            ;4227 ;
U 115F. 0003.0028.0180.0A80.0000.1326  ;4228  R[0]_NOT.0  ; Initial memory pattern (FFFFFFFF)
U 1326. 001B.001C.0580.0A88.0000.1327  ;4229  R[1]_NOT.K[.1]  ; Expected returned memory data
U 1327. 0018.0038.D180.0A98.0000.1160  ;4230  R[3]_K[.C0]  ; Load R[3] with the cache address of the check
            ;4231      ; bit patterns
U 1160. 0000.003D.0180.0800.0000.1162  ;4232 =0 CALL[TEST.35.TST] ; Execute the test
U 1161. 0000.003C.0180.0800.0000.1078  ;4233  J/RESTART.TEST.35 ; Restart this test for next controller
            ;4234  ;
            ;4235  ; The following code is the actual test and will be called as a
            ;4236  ; subroutine to prevent duplication of code.
            ;4237  ;
            ;4238 =0
            ;4239 TEST.35.TST:
U 1162. 0000.003D.0180.0800.0000.1234  ;4240  SUBTEST   ; Update the subtest counter
U 1163. 0800.003C.0180.0A00.0000.1328  ;4241  D_R[0]   ; Load D with initialize pattern and
U 1328. 0001.003C.0180.09C8.0000.1329  ;4242  RC[09]_D  ; save in RC[09]
U 1329. 0800.003C.0180.0A08.0000.132A  ;4243  D_R[1]   ; Load D with expected bit pattern and
U 132A. 0001.003C.0180.09E0.0000.132B  ;4244  RC[0C]_D  ; save in RC[0C]
U 132B. 0018.0038.7580.0A90.0000.132C  ;4245  R[2]_K[.20]  ; Set-up a counter for 32 patterns (passes)
            ;4246  ;
            ;4247  ; This point in the program is where we loop 32 times to run a bit
            ;4248  ; through a 32 bit longword.
            ;4249  ;
            ;4250 T.35.LOOP.POINT:
U 132C. 0010.0038.0180.0950.0200.132D  ;4251  VA_RC[0A]  ; Load VA with the memory test address
U 132D. 0810.0038.0180.0948.0000.132E  ;4252  D_RC[09]  ; Move init pattern to D for write
U 132E. 0000.003C.0180.A800.0000.1164  ;4253  CACHE.P_D[LONG]  ; Write the initialize pattern
U 1164. 0000.003D.0180.0800.0000.10BC  ;4254 =0 CALL[SBI.INIT]  ; Initialize the Sbi
U 1165. 0000.003C.0180.0800.0000.1166  ;4255  NOP
            ;4256  ;
            ;4257  ; Get the check bit pattern from cache and the do a SET.ECC by the
            ;4258  ; check bit pattern.
            ;4259  ;
U 1166. 0000.003D.0180.0800.0000.1257  ;4260 =0 CALL[ENABLE.CACHE] ; Turn on Cache
U 1167. 0000.003C.0180.0A18.0200.132F  ;4261  VA_R[3]   ; Load VA with cache address of check bit
            ;4262      ; patterns
U 132F. 0000.803C.0180.C800.0000.1330  ;4263  D[BYTE]_CACHE.P  ; Read one byte of cache data (byte = 1 check
            ;4264      ; bit pattern)
            ;4265  ALU_D.AND.K[.7F], ; Mask byte 0
U 1330. 0019.0034.5980.09C0.0000.1331  ;4266  RC[08]_ALU  ; Save in RC[08]
U 1331. 0800.003C.0180.0A18.0000.1332  ;4267  D_R[3]   ; Move cache address to the D register then
U 1332. 0019.0014.0580.0A98.0000.1168  ;4268  R[3]_D+K[.1]  ; update the pointer by 1
U 1168. 0000.003D.0180.0800.0000.1253  ;4269 =0 CALL[DISABLE.CACHE] ; Turn off Cache
U 1169. 0000.003C.0180.0800.0000.116A  ;4270  NOP
            ;4271 =0 D_RC[08],  ; Load D with the expected check bits and
U 116A. 0810.0039.0180.0940.0000.12B3  ;4272  CALL[SET_ECC]  ; Set the ECC bits according to the contents
            ;4273      ; of the D register
U 116B. 0000.003C.0180.0800.0000.116C  ;4274  NOP
            ;4275  ;
            ;4276  ; Now set-up diagnostic mode of 2 and read the data from memory do
            ;4277  ; force the error.
            ;4278  ;
U 116C. 0818.0039.0980.0800.0000.1247  ;4279 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode 2
U 116D. 0000.003C.0180.0800.0000.116E  ;4280  NOP
```

```
ZZ-ESKAR-V22  TEST 1C0 SINGLE BIT ERROR CORRECTION PART 2                                   SEQ 1313
ESKAR4B.MCR                    MICRO2  1P(00)    26-JUL-85  17:28:07                      Page  11

U 116E, 0000.003D.0180.0800.0000.122C  ;4281 =0 ERLOOP   ; Loop on error from here
U 116F, 0000.003C.0180.0800.0000.1170  ;4282  NOP
U 1170, 0000.003D.8580.0800.0084.726F  ;4283 =0 SET.TRAP.MASK[.C] ; Enable RDS Micro traps
U 1171, 0010.0038.0180.0950.0200.1333  ;4284  VA_RC[0A]  ; Load VA with the memory test address
U 1333, 0000.003C.0180.C800.0000.1334  ;4285  D[LONG]_CACHE.P  ; Read the location forcing a CRD error and
U 1334, 0001.003C.0180.09D8.0000.107C  ;4286  RC[0B]_D  ; save the read data in RC[0B]
           ;4287  ;
           ;4288  ; Now we have read the data, check to see if there was a RDS micro trap
           ;4289  ; and if there was then call an error as there shouldn't have been.
           ;4290  ;
U 107C, 0000.003D.0180.0800.0000.1269  ;4291 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 107D, 0000.003C.0180.0800.0000.1173  ;4292  J/T.35.CHK.DATA.1 ; No, continue with test
U 107E, 0000.003C.0180.0800.0000.1172  ;4293  J/T.35.RDS.UTRAP ; Yes, Call an error
           ;4294 =
```

```
          ;4295  .PAGE
          ;4296 ;
          ;4297 ;////////////////////////////////////////////////////////////////////////////////
          ;4298 ;
          ;4299 ; Got a RDS micro trap.
          ;4300 ; Call an error.
          ;4301 ;
          ;4302 =0
          ;4303 T.35.RDS.UTRAP:
U 1172, 0000,003D,0180,0800,0084,7054  ;4304  ERROR2[.8]
          ;4305 ;
          ;4306 ;////////////////////////////////////////////////////////////////////////////////
          ;4307 ;
          ;4308 ; ERROR LOGOUT:
          ;4309 ;
          ;4310 ; DATA: I/O Base Address of MS780-E Currently Under Test
          ;4311 ;  I/O Address of Configuration Register C/D
          ;4312 ;  Expected Memory Data
          ;4313 ;  Received Memory Data
          ;4314 ;  Memory Test Address
          ;4315 ;  Memory Initialize Pattern
          ;4316 ;  Expected Micro Trap Flags
          ;4317 ;  Received Micro Trap Flags
          ;4318 ;
          ;4319 ;////////////////////////////////////////////////////////////////////////////////
          ;4320 ;
          ;4321  ;
          ;4322  ; No RDS micro trap, now lets check the data we got while ECC was on
          ;4323  ; see if it is correct.
          ;4324  ;
          ;4325 T.35.CHK.DATA.1:
U 1173, 0810,0038,0180,0958,0000,1335  ;4326  D_RC[0B]  ; Load D with received data
          ;4327  ALU_D.XOR.RC[0C], ; Exclusive Or the received data with the
U 1335, 0011,0020,0180,0960,0010,1336  ;4328  CLK.UBCC  ; expected memory data
U 1336, 0000,013C,0180,0800,0000,1174  ;4329  Z?  ; Is the data correct ?
U 1174, 0000,003C,0180,0800,0000,1176  ;4330 =0 J/T.35.BAD.DATA.1 ; No, call an error
U 1175, 0000,003C,0180,0800,0000,1178  ;4331  J/T.35.DO.SEC.READ ; Yes, continue with the test
```

```
            ;4332  .PAGE
            ;4333 ;
            ;4334 ;////////////////////////////////////////////////////////////////////////////////
            ;4335 ;
            ;4336 ; Data received did not match data expected (ECC on ). (i.e., data
            ;4337 ; returned was not corrected)
            ;4338 ; Call an error.
            ;4339 ;
            ;4340 =0
            ;4341 T.35.BAD.DATA.1:
U 1176, 0000,003D,D580,0800,0084,7054  ;4342  ERROR2[.6]
U 1177, 0000,003C,0180,0800,0000,1178  ;4343  NOP
            ;4344 ;
            ;4345 ;////////////////////////////////////////////////////////////////////////////////
            ;4346 ;
            ;4347 ; ERROR LOGOUT:
            ;4348 ;
            ;4349 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4350 ;  I/O Address of Configuration Register C/D
            ;4351 ;  Expected Memory Data
            ;4352 ;  Received Memory Data
            ;4353 ;  Memory Test Address
            ;4354 ;  Memory Initialize Pattern
            ;4355 ;
            ;4356 ;////////////////////////////////////////////////////////////////////////////////
            ;4357 ;
            ;4358  ;
            ;4359  ; No errors with data while ECC on, lets turn ECC off and read the data
            ;4360  ; then check for data errors.
            ;4361  ;
            ;4362 =0
            ;4363 T.35.DO.SEC.READ:
U 1178, 0818,0039,0D80,0800,0000,1247  ;4364  SET.DIAG.MODE[.3] ; Select ECC Bypass mode
U 1179, 0010,0038,0180,0950,0200,1337  ;4365  VA_RC[0A]  ; Load VA with memory test address
U 1337, 0000,003C,0180,C800,0000,1338  ;4366  D[LONG]_CACHE.P  ; Read the data from the test address and
U 1338, 0001,003C,0180,09D8,0000,1339  ;4367  RC[0B]_D  ; save in RC[0B]
            ;4368  ALU_D.XOR.RC[0C], ; Exclusive or the received data with the
U 1339, 0011,0020,0180,0960,0010,133A  ;4369  CLK.UBCC  ; expected data.
U 133A, 0000,013C,0180,0800,0000,117A  ;4370  Z?  ; Does received data match expected data ?
U 117A, 0000,003C,0180,0800,0000,117C  ;4371 =0 J/T.35.BAD.DATA.2 ; No, Call an error
U 117B, 0000,003C,0180,0800,0000,133B  ;4372  J/T.35.CHK.R1  ; Yes, continue with test
```

```
            ;4373  .PAGE
            ;4374 ;
            ;4375 ;////////////////////////////////////////////////////////////////////////////
            ;4376 ;
            ;4377 ; Data received did not match data expected (ECC off ). (i.e., data was
            ;4378 ; NOT corrected on the array)
            ;4379 ; Call an error.
            ;4380 ;
            ;4381 =0
            ;4382 T.35.BAD.DATA.2:
U 117C, 0000.003D.D580.0800.0084.7054  ;4383  ERROR2[.6]
U 117D, 0000.003C.0180.0800.0000.133B  ;4384  NOP
            ;4385 ;
            ;4386 ;////////////////////////////////////////////////////////////////////////////
            ;4387 ;
            ;4388 ; ERROR LOGOUT:
            ;4389 ;
            ;4390 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4391 ;  I/O Address of Configuration Register C/D
            ;4392 ;  Expected Memory Data
            ;4393 ;  Received Memory Date
            ;4394 ;  Memory Test Address
            ;4395 ;  Memory Initialize Pattern
            ;4396 ;
            ;4397 ;////////////////////////////////////////////////////////////////////////////
            ;4398 ;
            ;4399  ;
            ;4400  ; Now check the value of the shift designator and if 0 then shift with
            ;4401  ; zero's in the low bit or if equal to one's then shift with a one in
            ;4402  ; the low bit.
            ;4403  ;
            ;4404 T.35.CHK.R1:
U 133B, 0000.003C.0180.0A00.0010.133C  ;4405  ALU_R[0],CLK.UBCC ; Load ALU with shift designator
U 133C, 0000.013C.0180.0800.0000.117E  ;4406  Z?   ; Is it 0 or one's ?
U 117E, 0000.003C.0180.0800.0000.133E  ;4407 =0 J/T.35.LEFT.WONES ; Is a 1, shift left with ones in
            ;4408  ;
            ;4409  ; Shift Left with Zero's in
            ;4410  ;
U 117F, 0810.0038.0180.0960.0000.133D  ;4411  D_RC[0C]  ; Is a 0, shift left with zero's in.  Load
            ;4412      ; pattern to be shifted into the D register
            ;4413  SI/MUL+,  ; Specify a shift in of Zero and
            ;4414  RC[0C]_D.LEFT,  ; perform the shift left one place
U 133D, 0021.003C.0300.09E0.0000.1340  ;4415  J/T.35.DECREMENT.R2 ; Now go decrement the pattern counter
            ;4416  ;
            ;4417  ; Shift Left with One's in
            ;4418  ;
            ;4419 T.35.LEFT.WONES:
U 133E, 0810.0038.0180.0960.0000.133F  ;4420  D_RC[0C]  ; Load pattern to shift into the D register
            ;4421  SI/MUL-,  ; Specify a shift in of One and
U 133F, 0021.003C.0380.09E0.0000.1340  ;4422  RC[0C]_D.LEFT  ; perform the shift left one place, then fall
            ;4423      ; through to decrement R[2].
            ;4424  ;
            ;4425  ; Decrement the pass count and if equal to 0 after the decrement then
            ;4426  ; return to the caller.
            ;4427  ;
```

```
            ;4428 T.35.DECREMENT.R2:
U 1340, 0800,003C,0180,0A10,0000,1341  ;4429  D_R[2]    ; Move pattern counter to the D register
            ;4430  ALU_D-K[.1],  ; Decrement the pattern counter,
            ;4431  R[2]_ALU,  ; restore back in R[2], and then
U 1341, 0019,0000,0580,0A90,0010,1342  ;4432  CLK.UBCC  ; clock condition codes to check value
U 1342, 0000,013C,0180,0800,0000,1180  ;4433  Z?    ; Is counter now equal to Zero ?
U 1180, 0000,003C,0180,0800,0000,132C  ;4434 =0 J/T.35.LOOP.POINT ; No, return to loop point for another pass
U 1181, 0000,003E,0180,0800,0000,0001  ;4435  RETURN1    ; Yes, return to the caller
            ;4436  ;
            ;4437  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4438  ;
            ;4439 =0
            ;4440 T.35.EXIT:
U 1182, 0000,003D,0180,0800,0000,12B9  ;4441  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
U 1183, 0000,003C,0180,0800,0000,1184  ;4442  NOP
            ;4443
```

```
;4444  .PAGE "TEST 1C1 SINGLE BIT ERROR CORRECTION PART 3"
;4445 ;************************************************************
;4446 ;+++
;4447 ;
;4448 ; Functional Description:
;4449 ; ---------------------
;4450 ;
;4451 ; This test verifies whether the ECC logic can detect and correct an
;4452 ; error in the check bits (in any bit position).  it initializes the
;4453 ; array location 0 to all zeroes.  One bit at a time the test comple-
;4454 ; ments the check bits for data of 0, and writes them to CNFG reg B.
;4455 ; In diagnostic mode 2, the data is read from the array.  It is
;4456 ; expected that the check bits will be corrected, the data returned
;4457 ; (and the data on the array) is still 0, the CRD bit in CNFG Reg C/D
;4458 ; is sets (bit 9), and that the correct syndromes are latched.
;4459 ;
;4460 ; The expected syndromes for this test will be generated by the test,
;4461 ; and they are as follows:
;4462 ;
;4463 ;  Check bit in error Syndrome
;4464 ;  ------------------ --------
;4465 ;
;4466 ;    Cx       01
;4467 ;    C0       02
;4468 ;    C1       04
;4469 ;    C2       08
;4470 ;    C4       10
;4471 ;    C8       20
;4472 ;    C16       40
;4473 ;
;4474 ;
;4475 ; Logic Description:
;4476 ; -----------------
;4477 ;  N/A
;4478 ;
;4479 ; Error Logout:
;4480 ; ------------
;4481 ;
;4482 ; See individual error reports.
;4483 ;
;4484 ; Sync Point Description:
;4485 ; ----------------------
;4486 ;  N/A
;4487 ;
;4488 ;============================================================
;4489 ;
;4490 ; Register Map:
;4491 ; ------------
;4492 ;
;4493 ;      RA,RB      Description:
;4494 ;      -----      -----------
;4495 ;
;4496 ; Not Used
;4497 ;
;4498 ;       RC        Description:
```

```
            ;4499 ;       --        ----------
            ;4500 ;
            ;4501 ;        E I/O Base Address of MS780-E Currently Under Test
            ;4502 ;
            ;4503 ;        D I/O Address of Configuration Register C/D
            ;4504 ;
            ;4505 ;        C Expected Syndrome Bits
            ;4506 ;
            ;4507 ;        B Received Syndrome Bits or Received Memory Data
            ;4508 ;
            ;4509 ;        A Expected Memory Data
            ;4510 ;
            ;4511 ;        9 Check Bit Pattern to be Written to Configuration Register B
            ;4512 ;
            ;4513 ; 6 Storage for the Contents of Configuration Register C/D
            ;4514 ;
            ;4515 ;--
            ;4516 ;*************************************************************
            ;4517 =0
U 1184, 0000,003D,0180,0800,0000,107F  ;4518 T.36: NEWTST   ; Signify start of new test
U 1185, 0000,003C,0180,0800,0000,1080  ;4519  NOP
U 1080, 0000,003D,0180,0800,0000,10C2  ;4520 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1081, 0000,003C,0180,0800,0000,11B8  ;4521  J/T.36.EXIT  ; No MS780-E's to test on sbi, exit this test
U 1082, 0000,003C,0180,0800,0000,1084  ;4522  NOP
            ;4523 =
            ;4524  ;
            ;4525  ; This is the entry point for the test to restart after finishing with
            ;4526  ; one controller.  The next controller is configured and the test
            ;4527  ; continues   If another controller is not present then the test is
            ;4528  ; exited.
            ;4529  ;
            ;4530 =00
            ;4531 RESTART.TEST.36:
U 1084, 0000,003D,0180,0800,0000,105C  ;4532  CALL[START.TEST] ; Configure the controllers
U 1085, 0000,003C,0180,0800,0000,11B8  ;4533  J/T.36.EXIT  ; No more controllers to configure
U 1086, 0018,0038,3180,09E0,0000,1343  ;4534  RC[0C]_K[.40]  ; Bit to be complemented in the Check Bits
            ;4535 =
U 1343, 0018,0038,3180,0998,0000,1186  ;4536  RC[3]_K[.40]  ; Also save in RC[3]
            ;4537  ;
            ;4538  ; This is the loop point for checking all seven syndrome bits.
            ;4539  ;
            ;4540 =0
            ;4541 T.36.LOOP.POINT:
U 1186, 0000,003D,0180,0800,0000,122C  ;4542  ERLOOP   ; Loop on error from here
U 1187, 0000,003C,0180,0800,0000,1188  ;4543  NOP
U 1188, 0000,003D,0180,0800,0000,10BC  ;4544 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1189, 0000,003C,0180,0800,0000,118A  ;4545  NOP
U 118A, 0000,003D,0180,0800,0000,12B9  ;4546 =0 CALL[MS780E.CLEANUP] ; Clear MS780-E/H configuration registers
U 118B, 0000,003C,0180,0800,0000,118C  ;4547  NOP
U 118C, 0000,003D,8580,0800,0084,726F  ;4548 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 118D, 0810,0038,0180,0960,0000,1344  ;4549  D_RC[0C]  ; Load D with the check bit complement bit then
            ;4550  ALU_D.XOR.K[.C], ; Form the check bit pattern to write to CNFG
U 1344, 0019,0020,8580,09C8,0000,1345  ;4551  RC[09]_ALU  ; register B
U 1345, 0010,0038,0180,0970,0200,1346  ;4552  VA_RC[0E]  ; Load VA with I/O Base Address of MS780-E
U 1346, 0000,003C,0180,0803,0000,1347  ;4553  VA_VA+4   ; Update VA to point to Configuration Reg B
```

```
U 1347, 0810.0038.0180.0948.0000.1348  ;4554  D_RC[09]  ; Load D with check bit pattern for write
U 1348, 0000.003C.0180.A800.0000.118E  ;4555  CACHE.P_D[LONG]  ; Write the check bit pattern to CNFG reg B
U 118E, 0818.0039.0980.0800.0000.1247  ;4556 =0 SET.DIAG.MODE[.2] ; Select Diagnostic mode 2
U 118F, 0F18.0038.1980.0800.0200.1349  ;4557  VA_K[ZERO],D_0  ; Point to location 0
U 1349, 0000.003C.0180.A800.0000.134A  ;4558  CACHE.P_D[LONG]  ; Initialize the memory location
U 134A, 0F00.003C.0180.0803.0000.134B  ;4559  VA_VA+4,D_0  ; Point tp location 4
U 134B, 0000.003C.0180.A800.0000.1190  ;4560  CACHE.P_D[LONG]  ; Initialize the memory location
U 1190, 0818.0039.D580.0800.0000.125B  ;4561 =0 D_K[.6],CALL[WAIT.LOOP] ; Wait for the writes to finish
U 1191, 0018.0038.1980.0800.0200.134C  ;4562  VA_K[ZERO]  ; Point to location 0
U 134C, 0000.003C.0180.C800.0000.134D  ;4563  D[LONG]_CACHE.P  ; Read the location to force the CRD then
U 134D, 0001.003C.0180.09D0.0000.1088  ;4564  RC[0A]_D  ; save the read data in RC[0A]
U 1088, 0000.003D.0180.0800.0000.1269  ;4565 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 1089, 0000.003C.0180.0800.0000.1193  ;4566  J/T.36.CHK.BIT9  ; No, continue with the test
U 108A, 0000.003C.0180.0800.0000.1192  ;4567  J/T.36.RDS.UTRAP.1 ; Yes, Call an error
          ;4568 =
```

```
            ;4569  .PAGE
            ;4570 ;
            ;4571 ;////////////////////////////////////////////////////////////////////////////////
            ;4572 ;
            ;4573 ; Got a RDS micro trap.
            ;4574 ; Call an error
            ;4575 ;
            ;4576 =0
            ;4577 T.36.RDS.UTRAP.1:
U 1192, 0000,003D,0180,0800,0084,7054  ;4578  ERROR2[.8]
            ;4579 ;
            ;4580 ;////////////////////////////////////////////////////////////////////////////////
            ;4581 ;
            ;4582 ; ERROR LOGOUT:
            ;4583 ;
            ;4584 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4585 ;  I/O Address of Configuration Register C/D
            ;4586 ;  NOT USED
            ;4587 ;  NOT USED
            ;4588 ;  NOT USED
            ;4589 ;  Check Bit Pattern Written to Configuration Register B
            ;4590 ;  Expected Micro Trap Flags
            ;4591 ;  Received Micro Trap Flags
            ;4592 ;
            ;4593 ;////////////////////////////////////////////////////////////////////////////////
            ;4594 ;
            ;4595 T.36.CHK.BIT9:
U 1193, 0010,0038,0180,0968,0200,134E  ;4596  VA_RC[0D]  ; Load VA with address of Configuration reg C/D
U 134E, 0000,003C,0180,C800,0000,134F  ;4597  D[LONG]_CACHE.P  ; Read the contents of CNFG C/D
U 134F, 0001,003C,0180,09B0,0000,1350  ;4598  RC[06]_D  ; Save CNFG C/D data in RC[06]
U 1350, 0000,003C,D980,0800,0084,7351  ;4599  SC_K[.9]  ; Get ready to mask bit 9 from CNFG C/D
U 1351, 0003,001C,0180,09C0,0000,1352  ;4600  RC[8]_NOT.MASK  ; Expected status of Bit 9 in CNFG Reg C/D
            ;4601  D_D.ANDNOT.MASK, ; Mask out bit 9 from Configuration Register
U 1352, 0801,0024,0180,0800,0010,1353  ;4602  CLK.UBCC ; ...C/D and check to see if set
U 1353, 0001,013C,0180,09B8,0000,1194  ;4603  Z?,RC[7]_D  ; Was Bit 9 set ?
U 1194, 0000,003C,0180,0800,0000,1197  ;4604 =0 J/T.36.CHK.SYNDROME ; Yes, continue testing
U 1195, 0000,003C,0180,0800,0000,1196  ;4605  NOP  ; No, should have been, call an error
```

```
            ;4606  .PAGE
            ;4607 ;
            ;4608 ;////////////////////////////////////////////////////////////////////////////////
            ;4609 ;
            ;4610 ; Configuration Register C/D bit 9 ( CRD Flag ) was NOT set.
            ;4611 ; Call an error.
            ;4612 ;
            ;4613 =0
U 1196, 0000.003D.D980.0800.0084.7054  ;4614  ERROR2[.9]
            ;4615 ;
            ;4616 ;////////////////////////////////////////////////////////////////////////////////
            ;4617 ;
            ;4618 ; ERROR LOGOUT:
            ;4619 ;
            ;4620 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4621 ;  I/O Address of Configuration Register C/D
            ;4622 ;  Not Used
            ;4623 ;  Not Used
            ;4624 ;  Not Used
            ;4625 ;  Check Bit Pattern Written to Configuration Register B
            ;4626 ;  Expected bit to be set in CNFG Reg C/D
            ;4627 ;  Received status of the bit in CNFG Reg C/D
            ;4628 ;  Contents of CNFG Reg C/D
            ;4629 ;
            ;4630 ;////////////////////////////////////////////////////////////////////////////////
            ;4631 ;
            ;4632 T.36.CHK.SYNDROME:
U 1197, 0810.0038.0180.0930.0000.1354  ;4633  D_RC[06]  ; Move Saved Configuration Reg C/D to D reg
            ;4634  ALU_D.AND.K[.7F], ; Mask out bits <06:00> (syndrome bits) and
U 1354, 0019.0034.5980.09D8.0000.1355  ;4635  RC[0B]_ALU  ; save in RC[0B]
U 1355, 0810.0038.0180.0960.0000.1356  ;4636  D_RC[0C]  ; Move expected syndrome bits to the D reg
            ;4637  ALU_D.XOR.RC[0B], ; Exclusive Or the expected syndrome bits with
U 1356, 0011.0020.0180.0958.0010.1357  ;4638  CLK.UBCC  ; the received syndrome bits
U 1357, 0000.013C.0180.0800.0000.1198  ;4639  Z?  ; Are they equal ?
U 1198, 0000.003C.0180.0800.0000.119A  ;4640 =0 J/T.36.SYNDROME.ERROR ; No, Call an error
U 1199, 0000.003C.0180.0800.0000.119B  ;4641  J/T.36.DO.NEXT  ; Yes, continue with test
```

```
            ;4642  .PAGE
            ;4643 ;
            ;4644 ;////////////////////////////////////////////////////////////////////////////
            ;4645 ;
            ;4646 ; Received and Expected syndrome bits did not match. Call an error.
            ;4647 ;
            ;4648 =0
            ;4649 T.36.SYNDROME.ERROR:
U 119A, 0000,003D,D580,0800,0084,7054  ;4650  ERROR2[.6]
            ;4651 ;
            ;4652 ;////////////////////////////////////////////////////////////////////////////
            ;4653 ;
            ;4654 ; ERROR LOGOUT:
            ;4655 ;
            ;4656 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4657 ;  I/O Address of Configuration Register C/D
            ;4658 ;  Expected Syndrome Bits
            ;4659 ;  Received Syndrome Bits
            ;4660 ;  Not Used
            ;4661 ;  Check Bit Pattern Written to Configuration Register B
            ;4662 ;
            ;4663 ;////////////////////////////////////////////////////////////////////////////
            ;4664 ;
            ;4665 T.36.DO.NEXT:
U 119B, 0003,003C,0180,09E0,0000,119C  ;4666  RC[0C]_0  ; Expected data from memory after forcing
            ;4667      ; the CRD error
U 119C, 0000,003D,0180,0800,0000,122C  ;4668 =0 ERLOOP   ; Loop on error from here
U 119D, 0000,003C,0180,0800,0000,119E  ;4669  NOP
U 119E, 0000,003D,0180,0800,0000,10BC  ;4670 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 119F, 0010,0038,0180,0970,0200,1358  ;4671  VA_RC[0E]  ; Point to CNFG reg B
            ;4672  VA_VA+4,  ; ...
U 1358, 0810,0038,0180,094B,0000,1359  ;4673   D_RC[9]  ; Get Checkbits for test
U 1359, 0000,003C,0180,A800,0000,11A0  ;4674  CACHE.P_D[LONG]  ; Initialize the Substitue ECC bits in CNFG
            ;4675      ; ...Reg B
U 11A0, 0818,0039,0980,0800,0000,1247  ;4676 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode 2
U 11A1, 0000,003C,0180,0800,0000,11A2  ;4677  NOP
U 11A2, 0000,003D,8580,0800,0084,726F  ;4678 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
            ;4679  VA_K[ZERO],  ; Point VA to location 0
U 11A3, 0F18,0038,1980,0800,0200,135A  ;4680   D_0  ; Zero D register for writing to memory
U 135A, 0000,003C,0180,A800,0000,135B  ;4681  CACHE.P_D[LONG]  ; Initialize memory to 0
U 135B, 0000,003C,0180,0800,0000,135C  ;4682  NOP
U 135C, 0000,003C,0180,C800,0000,135D  ;4683  D[LONG]_CACHE.P  ; Read the array and force the CRD
U 135D, 0001,003C,0180,09D8,0000,108C  ;4684  RC[0B]_D  ; Store the read data in RC[0A]
U 108C, 0000,003D,0180,0800,0000,1269  ;4685 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 108D, 0000,003C,0180,0800,0000,11A5  ;4686  J/T.36.CHK.DATA  ; No, continue with the test
U 108E, 0000,003C,0180,0800,0000,11A4  ;4687  J/T.36.RDS.UTRAP.2 ; Yes, call an error
            ;4688 =
```

```
            ;4689  .PAGE
            ;4690 ;
            ;4691 ;////////////////////////////////////////////////////////////////////////////////
            ;4692 ;
            ;4693 ; Got a RDS micro trap when reading back the data.
            ;4694 ; Call an error.
            ;4695 ;
            ;4696 =0
            ;4697 T.36.RDS.UTRAP.2:
U 11A4, 0000,003D,0180,0800,0084,7054  ;4698  ERROR2[.8]
            ;4699 ;
            ;4700 ;////////////////////////////////////////////////////////////////////////////////
            ;4701 ;
            ;4702 ; ERROR LOGOUT:
            ;4703 ;
            ;4704 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4705 ;  I/O Address of Configuration Register C/D
            ;4706 ;  Not Used
            ;4707 ;  Not Used
            ;4708 ;  Not Used
            ;4709 ;  Check Bit Pattern Written to Configuration Register B
            ;4710 ;  Expected Micro Trap Flags
            ;4711 ;  Received Micro Trap Flags
            ;4712 ;
            ;4713 ;////////////////////////////////////////////////////////////////////////////////
            ;4714 ;
            ;4715 T.36.CHK.DATA:
U 11A5, 0810,0038,0180,0960,0000,135E  ;4716  D_RC[0C]  ; Load D with the expected memory data
            ;4717  ALU_D.XOR.RC[0B], ; Exclusive Or the expected memory data with
U 135E, 0011,0020,0180,0958,0010,135F  ;4718   CLK.UBCC ; ...received memory data.
U 135F, 0000,013C,0180,0800,0000,11A6  ;4719  Z?   ; Were they equal ?
U 11A6, 0000,003C,0180,0800,0000,11A8  ;4720 =0 J/T.36.BAD.DATA.1 ; No, Call an error
U 11A7, 0000,003C,0180,0800,0000,11AA  ;4721  J/T.36.DO.SEC.READ ; Yes, continue with test
```

```
            ;4722  .PAGE
            ;4723 ;
            ;4724 ;////////////////////////////////////////////////////////////////////////////
            ;4725 ;
            ;4726 ; Received and Expected data did not match (with ECC on).
            ;4727 ; (i.e., Memory modified the data returned but the error was in the checkbits)
            ;4728 ; Call an error.
            ;4729 ;
            ;4730 =0
            ;4731 T.36.BAD.DATA.1:
U 11A8, 0000,003D,D580,0800,0084,7054  ;4732  ERROR2[.6]
U 11A9, 0000,003C,0180,0800,0000,11AA  ;4733  NOP
            ;4734 ;
            ;4735 ;////////////////////////////////////////////////////////////////////////////
            ;4736 ;
            ;4737 ; ERROR LOGOUT:
            ;4738 ;
            ;4739 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4740 ;  I/O Address of Configuration Register C/D
            ;4741 ;  Expected memory Data
            ;4742 ;  Received memory Data
            ;4743 ;  Not Used
            ;4744 ;  Check Bit Pattern Written to Configuration Register B
            ;4745 ;
            ;4746 ;////////////////////////////////////////////////////////////////////////////
            ;4747 ;
            ;4748 =0
            ;4749 T.36.DO.SEC.READ:
U 11AA, 0818,0039,0D80,0800,0000,1247  ;4750  SET.DIAG.MODE[.3] ; Disable ECC logic
U 11AB, 0018,0038,1980,0800,0200,1360  ;4751  VA_K[ZERO]  ; Point VA to the test location
U 1360, 0000,003C,0180,C800,0000,1361  ;4752  D[LONG]_CACHE.P  ; Read the data from the test location and
U 1361, 0001,003C,0180,09D8,0000,1362  ;4753  RC[0B]_D  ; ...save in RC[0B]
            ;4754  ALU_D.XOR.RC[0C], ; Exclusive Or Received data and the Expected
U 1362, 0011,0020,0180,0960,0010,1363  ;4755  CLK.UBCC  ; memory data
U 1363, 0000,013C,0180,0800,0000,11AC  ;4756  Z?  ; Were they equal ?
U 11AC, 0000,003C,0180,0800,0000,11AE  ;4757 =0 J/T.36.BAD.DATA.2 ; No, Call an error
U 11AD, 0000,003C,0180,0800,0000,11B0  ;4758  J/T.36.DO.NEXT.1 ; Yes, continue with the test
            ;4759 ;
            ;4760 ;////////////////////////////////////////////////////////////////////////////
            ;4761 ;
            ;4762 ; Received and Expected data did not match (with ECC off).
            ;4763 ; (i.e.,Memory modified the data on the array)
            ;4764 ; Call an error.
            ;4765 ;
            ;4766 =0
            ;4767 T.36.BAD.DATA.2:
U 11AE, 0000,003D,D580,0800,0084,7054  ;4768  ERROR2[.6]
U 11AF, 0000,003C,0180,0800,0000,11B0  ;4769  NOP
            ;4770 ;
            ;4771 ;////////////////////////////////////////////////////////////////////////////
            ;4772 ;
            ;4773 ; ERROR LOGOUT:
            ;4774 ;
            ;4775 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4776 ;  I/O Address of Configuration Register C/D
```

```
            ;4777 ;  Expected Memory Data
            ;4778 ;  Received Memory Data
            ;4779 ;  Not Used
            ;4780 ;  Check Bit Pattern Written to Configuration Register B
            ;4781 ;
            ;4782 ;////////////////////////////////////////////////////////////////////////////////
            ;4783 ;
            ;4784 =0
            ;4785 T.36.DO.NEXT.1:
U 11B0, 0818.0039.0D80.0800.0000.1247  ;4786  SET.DIAG.MODE[.3] ; Set ECC bypass mode
U 11B1, 0018.0038.1980.0800.0200.1364  ;4787  VA_K[ZERO]  ; Point VA to test location 0
U 1364, 0000.003C.0180.C800.0000.1365  ;4788  D[LONG]_CACHE.P  ; Read the test location
U 1365, 0010.0038.0180.0968.0200.1366  ;4789  VA_RC[0D]  ; Load VA with address of CNFG C/D
U 1366, 0000.003C.0180.C800.0000.1367  ;4790  D[LONG]_CACHE.P  ; Read CNFG register C/D
U 1367, 0019.0034.5980.09E0.0000.1368  ;4791  RC[0C]_D.AND.K[.7F] ; Mask out bits <06:00> from CNFG C/D
U 1368, 0018.0038.8580.09D8.0000.1369  ;4792  RC[0B]_K[.C]  ; Expected Check bits
U 1369, 0810.0038.0180.0958.0000.136A  ;4793  D_RC[0B]  ; Load D with expected check bits for compare
            ;4794  ALU_D.XOR.RC[0C], ; Exclusive Or the expected data with the
U 136A, 0011.0020.0180.0960.0010.136B  ;4795  CLK.UBCC  ; received check bits.
U 136B, 0000.013C.0180.0800.0000.11B2  ;4796  Z?  ; Were they equal ?
U 11B2, 0000.003C.0180.0800.0000.11B4  ;4797 =0 J/T.36.CHK.BIT.ERROR ; No, Call an error
U 11B3, 0000.003C.0180.0800.0000.11B5  ;4798  J/T.36.SHIFT.RC.0B ; Yes, continue with the test
```

```
            ;4799  .PAGE
            ;4800 ;
            ;4801 ;////////////////////////////////////////////////////////////////////////////
            ;4802 ;
            ;4803 ; Received and Expected check bits did not match. (i.e., checkbits were
            ;4804 ; NOT corrected)
            ;4805 ; Call an error.
            ;4806 ;
            ;4807 =0
            ;4808 T.36.CHK.BIT.ERROR:
U 11B4, 0000,003D,D580,0800,0084,7054  ;4809  ERROR2[.6]
            ;4810 ;
            ;4811 ;////////////////////////////////////////////////////////////////////////////
            ;4812 ;
            ;4813 ; ERROR LOGOUT:
            ;4814 ;
            ;4815 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4816 ;  I/O Address of Configuration Register C/D
            ;4817 ;  Expected Checkbits
            ;4818 ;  Received Checkbits
            ;4819 ;  Not Used
            ;4820 ;  Check Bit Pattern Written to Configuration Register B
            ;4821 ;
            ;4822 ;////////////////////////////////////////////////////////////////////////////
            ;4823 ;
            ;4824 T.36.SHIFT.RC.0B:
U 11B5, 0810,0038,0180,0918,0000,136C  ;4825  D_RC[3]   ; Get ready to shift Expected syndrome pattern
U 136C, 0600,003C,0180,0800,0000,136D  ;4826  SI/ZERO,D_D.RIGHT ; Shift it right with a zero in
U 136D, 0001,003C,0180,09E0,0000,136E  ;4827  RC[0C]_D  ; Restore shifted expected syndrome pattern
            ;4828  RC[3]_D,  ; ...
U 136E, 0001,003C,0180,0998,0010,136F  ;4829   CLK.UBCC ; Check the value of expected syndrome pattern
            ;4830      ; and if it is equal to 0 then restart test
U 136F, 0000,013C,0180,0800,0000,11B6  ;4831  Z?   ; Is it zero ?
U 11B6, 0000,003C,0180,0800,0000,1186  ;4832 =0 J/T.36.LOOP.POINT ; No, continue with this controller
U 11B7, 0000,003C,0180,0800,0000,1084  ;4833  J/RESTART.TEST.36 ; Yes, restart this test for next controller
            ;4834  ;
            ;4835  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!
            ;4836  ;
            ;4837 =0
            ;4838 T.36.EXIT:
U 11B8, 0000,003D,0180,0800,0000,12B9  ;4839  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
U 11B9, 0000,003C,0180,0800,0000,11BA  ;4840  NOP
            ;4841
```

```
;4842  .PAGE "TEST 1C2 DOUBLE BIT ERROR TEST PART 1"
;4843 ;*************************************************************
;4844 ;+++
;4845 ;
;4846 ; Functional Description:
;4847 ; ----------------------
;4848 ;
;4849 ; This test forces double bit errors and verifies that they are detected.
;4850 ; It makes use of Diagnostic mode 2 and errors are forced in two bits of
;4851 ; location 0 on each Controller.  The test passes only if the following
;4852 ; are true: RDS tag is sent on the SBI when a double bit error occurrs,
;4853 ; bits 28 (Error Log Request) and 10 (RDS error) in CNFG C (or D)
;4854 ; register are set, the data is unmodified on the array and the error
;4855 ; syndrome contains an even number of ones (ie., the ECC logic detects
;4856 ; a RDS).
;4857 ;
;4858 ; The test will float two ones in a field of zeroes through location 0
;4859 ; while forcing RDS errors by setting the ECC substitute bits in CNFG
;4860 ; Register B to xC (for data of all zeroes).
;4861 ;
;4862 ;
;4863 ; Logic Description:
;4864 ; -----------------
;4865 ; N/A
;4866 ;
;4867 ; Error Logout:
;4868 ; ------------
;4869 ;
;4870 ; See individual error reports.
;4871 ;
;4872 ; Sync Point Description:
;4873 ; ----------------------
;4874 ; N/A
;4875 ;
;4876 ;=============================================================
;4877 ;
;4878 ; Register Map:
;4879 ; ------------
;4880 ;
;4881 ;      RA,RB      Description:
;4882 ;      -----      -----------
;4883 ;
;4884 ;        0        Pattern Counter
;4885 ;
;4886 ;       RC        Description:
;4887 ;       --        -----------
;4888 ;
;4889 ;        E I/O Base Address of MS780-E Currently Under Test
;4890 ;
;4891 ;        D I/O Address of Configuration Register C/D
;4892 ;
;4893 ;        C Data Pattern Under Test/Expected Memory Data
;4894 ;
;4895 ;        B Received Memory Data
;4896 ;
```

```
            ;4897 ;       A Contents of Configuration Register C/D
            ;4898 ;
            ;4899 ;--
            ;4900 ;*************************************************************
            ;4901 =0
U 11BA, 0000,003D,0180,0800,0000,107F  ;4902 T.37: NEWTST   ; Signify start of new test
U 11BB, 0000,003C,0180,0800,0000,1090  ;4903  NOP
U 1090, 0000,003D,0180,0800,0000,10C2  ;4904 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1091, 0000,003C,0180,0800,0000,11EA  ;4905  J/T.37.EXIT  ; No MS780-E's to test, exit
U 1092, 0000,003C,0180,0800,0000,1094  ;4906  NOP
            ;4907 =
            ;4908  ;
            ;4909  ; This is the point where the test is restarted to test the next
            ;4910  ; controller that is on the SBI.  After each controller is completely
            ;4911  ; tested, control will be returned back to this point where the next
            ;4912  ; controller is configured and tested.  If there is no more controllers
            ;4913  ; on the SBI then the test will be exited.
            ;4914  ;
            ;4915 =00
            ;4916 RESTART.TEST.37:
U 1094, 0000,003D,0180,0800,0000,105C  ;4917  CALL[START.TEST] ; Configure the Controllers
U 1095, 0000,003C,0180,0800,0000,11EA  ;4918  J/T.37.EXIT  ; No more controllers to configure
U 1096, 0018,0038,0D80,09E0,0000,1097  ;4919  RC[0C]_K[.3]  ; Load RC[0C] with data pattern used to force
            ;4920      ; RDS error
U 1097, 0018,0038,8D80,0A80,0000,11BC  ;4921  R[0]_K[.1F]  ; Set up Pattern counter
            ;4922  ;
            ;4923  ; This is the point where the loop for 31 times executed.  The test
            ;4924  ; will float a pair of ones thourgh a 32 bit longword by shifting
            ;4925  ; the pair left one place (with zero in), decrement the pattern counter
            ;4926  ; and if pattern counter is not equal to zero then the test Jumps back
            ;4927  ; to this point for another pass.
            ;4928  ;
            ;4929 =0
            ;4930 T.37.LOOP.POINT:
U 11BC, 0000,003D,0180,0800,0000,122C  ;4931  ERLOOP   ; Loop on error from here
U 11BD, 0000,003C,0180,0800,0000,11BE  ;4932  NOP
U 11BE, 0000,003D,0180,0800,0000,10BC  ;4933 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11BF, 0000,003C,0180,0800,0000,11C0  ;4934  NOP
U 11C0, 0000,003D,0180,0800,0000,12B9  ;4935 =0 CALL[MS780E.CLEANUP] ; Clear Memory Configuration Registers
U 11C1, 0000,003C,0180,0800,0000,11C2  ;4936  NOP
U 11C2, 0818,0039,0980,0800,0000,1247  ;4937 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode 2
U 11C3, 0000,003C,0180,0800,0000,11C4  ;4938  NOP
U 11C4, 0818,0039,8580,0800,0000,12B3  ;4939 =0 SET.ECC[.C]  ; Set ECC Substitute bits in CNFG reg B to xC
U 11C5, 0018,0038,1980,0800,0200,1370  ;4940  VA_K[ZERO]  ; Point VA to test location
U 1370, 0810,0038,0180,0960,0000,1371  ;4941  D_RC[0C]  ; Load D with the test data to be written
U 1371, 0000,003C,0180,A800,0000,11C6  ;4942  CACHE.P_D[LONG]  ; Write the test pattern
U 11C6, 0000,003D,8580,0800,0084,726F  ;4943 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 11C7, 0000,003C,0180,C800,0000,1098  ;4944  D[LONG]_CACHE.P  ; Read the location under test
            ;4945  ;
            ;4946  ; Check to see if there was a micro trap on the Read that forced the
            ;4947  ; error.  There should have been a RDS micro trap and if there wasn't
            ;4948  ; then call an error.
            ;4949  ;
U 1098, 0000,003D,0180,0800,0000,1269  ;4950 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 1099, 0000,003C,0180,0800,0000,11C8  ;4951  J/T.37.GOT.NO.RDS.UTRAP ; No, should have been, call an error
```

```
U 109A, 0000,003C,0180,0800,0000,11C9  ;4952  J/T.37.CHK.BIT10 ; Yes, should have been, continue with test
            ;4953 =
```

```
            ;4954  .PAGE
            ;4955 ;
            ;4956 ;////////////////////////////////////////////////////////////////////////////////
            ;4957 ;
            ;4958 ; Didn't get a RDS micro trap when reading back the data.
            ;4959 ; (i.e., the RDS error was NOT detected)
            ;4960 ; Call an error.
            ;4961 ;
            ;4962 =0
            ;4963 T.37.GOT.NO.RDS.UTRAP:
U 11C8, 0000,003D,0180,0800,0084,7054  ;4964  ERROR2[.8]
            ;4965 ;
            ;4966 ;////////////////////////////////////////////////////////////////////////////////
            ;4967 ;
            ;4968 ; ERROR LOGOUT:
            ;4969 ;
            ;4970 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4971 ;  I/O Address of Configuration Register C/D
            ;4972 ;  Data Pattern Under Test
            ;4973 ;  Unused
            ;4974 ;  Unused
            ;4975 ;  Unused
            ;4976 ;  Expected Micro Trap Flags
            ;4977 ;  Received Micro Trap Flags
            ;4978 ;
            ;4979 ;////////////////////////////////////////////////////////////////////////////////
            ;4980 ;
            ;4981  ;
            ;4982  ; Now check bit 10 (RDS error) and insure that it got set.  If it
            ;4983  ; didn't then call an error
            ;4984  ;
            ;4985 T.37.CHK.BIT10:
U 11C9, 0010,0038,0180,0968,0200,1372  ;4986  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 1372, 0000,003C,0180,C800,0000,1373  ;4987  D[LONG]_CACHE.P  ; Read the contents of CNFG  C/D
U 1373, 0001,003C,0180,09C8,0000,1374  ;4988  RC[9]_D  ; Save CNFG C/D data in RC[9]
U 1374, 0000,003C,F580,0800,0084,7375  ;4989  SC_K[.A]  ; Get ready to mask Bit 10
U 1375, 0003,001C,0180,09D8,0000,1376  ;4990  RC[0B]_NOT.MASK  ; Expected bit to be set in CNFG Reg C/D
            ;4991  D_D.ANDNOT.MASK, ; Mask out bit 10
U 1376, 0801,0024,0180,0800,0010,1377  ;4992  CLK.UBCC ; Check for Bit 10 set
U 1377, 0001,013C,0180,09D0,0000,11CA  ;4993  Z?,RC[0A]_D  ; Was bit 10 set ?
U 11CA, 0000,003C,0180,0800,0000,11CD  ;4994 =0 J/T.37.CHK.BIT28 ; Yes, continue with test
U 11CB, 0000,003C,0180,0800,0000,11CC  ;4995  NOP  ; NO, call an error
```

```
            ;4996  .PAGE
            ;4997 ;
            ;4998 ;////////////////////////////////////////////////////////////////////////////
            ;4999 ;
            ;5000 ; Configuration Register bit 10 ( RDS Flag ) was not set.
            ;5001 ; (i.e., RDS error was NOT detected by the error register)
            ;5002 ; Call an error.
            ;5003 ;
            ;5004 =0
U 11CC, 0000,003D,D580,0800,0084,7054  ;5005  ERROR2[.6]
            ;5006 ;
            ;5007 ;////////////////////////////////////////////////////////////////////////////
            ;5008 ;
            ;5009 ; ERROR LOGOUT:
            ;5010 ;
            ;5011 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5012 ;  I/O Address of Configuration Register C/D
            ;5013 ;  Data Pattern Under Test/Expected Data Pattern
            ;5014 ;  Expected bit to be set in CNFG Reg C/D
            ;5015 ;  Received Status of the bit in CNFG Reg C/D
            ;5016 ;  Contents of Configuration Register C/D
            ;5017 ;
            ;5018 ;////////////////////////////////////////////////////////////////////////////
            ;5019 ;
            ;5020  ;
            ;5021  ; Now check Bit 28 ( Error log request) and insure that it was set.
            ;5022  ; If it wasn't then call an error.
            ;5023  ;
            ;5024 T.37.CHK.BIT28:
U 11CD, 001B,0010,ED80,0800,0082,1378  ;5025  SC_K[.1B]+1  ; Get ready to mask Bit 28
U 1378, 0003,001C,0180,C9D8,0000,1379  ;5026  RC[0B]_NOT.MASK  ; Bit expected to be set in CNFG Reg C/D
U 1379, 0810,0038,0180,0948,0000,137A  ;5027  D_RC[9]  ; Load saved CNFG C/D into the D register
            ;5028  D_D.ANDNOT.MASK, ; Mask out bit 28
U 137A, 0801,0024,0180,0800,0010,137B  ;5029   CLK.UBCC ; Check for Bit 28 set
U 137B, 0001,013C,0180,09D0,0000,11CE  ;5030  Z?,RC[0A]_D  ; Was bit 28 set ?
U 11CE, 0000,003C,0180,0800,0000,11D2  ;5031 =0 J/T.37.DO.NEXT.1 ; Yes, continue with test
U 11CF, 0000,003C,0180,0800,0000,11D0  ;5032  NOP  ; NO, call an error
```

```
            ;5033  .PAGE
            ;5034  ;
            ;5035  ;////////////////////////////////////////////////////////////////////////////
            ;5036  ;
            ;5037  ; Coniguration Register bit 28 ( Error Log Request ) was not set.
            ;5038  ; Call an error.
            ;5039  ;
            ;5040  =0
U 11D0. 0000.003D.D580.0800.0084.7054  ;5041  ERROR2[.6]
U 11D1. 0000.003C.0180.0800.0000.11D2  ;5042  NOP
            ;5043  ;
            ;5044  ;////////////////////////////////////////////////////////////////////////////
            ;5045  ;
            ;5046  ; ERROR LOGOUT:
            ;5047  ;
            ;5048  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5049  ;  I/O Address of Configuration Register C/D
            ;5050  ;  Data Pattern Under Test/Expected Data Pattern
            ;5051  ;  Expected bit to be set in CNFg Reg C/D
            ;5052  ;  Received Status of the bit in CNFG Reg C/D
            ;5053  ;  Contents of Configuration Register C/D
            ;5054  ;
            ;5055  ;////////////////////////////////////////////////////////////////////////////
            ;5056  ;
            ;5057   ;
            ;5058   ; Ok now lets read the location again and this time check the syndrome
            ;5059   ; bits, the data in the memory location , and the SBI error register.
            ;5060   ;
            ;5061  =0
            ;5062  T.37.DO.NEXT.1:
U 11D2. 0000.003D.0180.0800.0000.122C  ;5063   ERLOOP    ; Loop on error from here
U 11D3. 0000.003C.0180.0800.0000.11D4  ;5064   NOP
U 11D4. 0000.003D.0180.0800.0000.10BC  ;5065  =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11D5. 0000.003C.0180.0800.0000.11D6  ;5066   NOP
U 11D6. 0818.0039.0980.0800.0000.1247  ;5067  =0 SET.DIAG.MODE[.2] ; Set diagnostic mode 2
U 11D7. 0000.003C.0180.0800.0000.11D8  ;5068   NOP
U 11D8. 0818.0039.8580.0800.0000.12B3  ;5069  =0 SET.ECC[.C]  ; Set ECC substitute bits in CNFG Reg B to xC
U 11D9. 0018.0038.1980.0800.0200.137C  ;5070   VA_K[ZERO]  ; Point VA to location 0
U 137C. 0810.0038.0180.0960.0000.137D  ;5071   D_RC[0C]  ; Load D with test pattern
U 137D. 0000.003C.0180.A800.0000.11DA  ;5072   CACHE.P_D[LONG]  ; Write the test pattern
U 11DA. 0000.003D.8580.0800.0084.726F  ;5073  =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 11DB. 0000.003C.0180.C800.0000.109C  ;5074   D[LONG]_CACHE.P  ; Read the data from the test location
            ;5075   ;
            ;5076   ; Check to see if there was a RDS micro trap and if not then call
            ;5077   ; an error
            ;5078   ;
U 109C. 0000.003D.0180.0800.0000.1269  ;5079  =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 109D. 0000.003C.0180.0800.0000.11DC  ;5080   J/T.37.GOT.NO.UTRAP.27 ; No, should have got one, call an error
U 109E. 0000.003C.0180.0800.0000.10A0  ;5081   J/T.37.CHK.SBIERR ; Yes, continue testing
            ;5082  =
```

```
            ;5083  .PAGE
            ;5084 ;
            ;5085 ;////////////////////////////////////////////////////////////////////////////////
            ;5086 ;
            ;5087 ; Didn't get a RDS micro trap on the read.
            ;5088 ; Call an error.
            ;5089 ;
            ;5090 =0
            ;5091 T.37.GOT.NO.UTRAP.27:
U 11DC. 0000.003D.0180.0800.0084.7054  ;5092  ERROR2[.8]
U 11DD. 0000.003C.0180.0800.0000.10A0  ;5093  NOP
            ;5094 ;
            ;5095 ;////////////////////////////////////////////////////////////////////////////////
            ;5096 ;
            ;5097 ; ERROR LOGOUT:
            ;5098 ;
            ;5099 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5100 ;  I/O Address of Configuration Register C/D
            ;5101 ;  Data Pattern Under Test
            ;5102 ;  Not Used
            ;5103 ;  Not Used
            ;5104 ;  Not Used
            ;5105 ;  Expected Micro Trap Flags
            ;5106 ;  Received Micro Trap Flags
            ;5107 ;
            ;5108 ;////////////////////////////////////////////////////////////////////////////////
            ;5109 ;
            ;5110  ;
            ;5111  ; Check the SBI error register to insure the fact that the CPU
            ;5112  ; received the RDS tag from the memory
            ;5113  ;
            ;5114 =00
            ;5115 T.37.CHK.SBIERR:
U 10A0. 0000.003D.8980.0800.0084.72C9  ;5116  CHECK.SBI.ERR[.D] ; Did the CPU log the RDS error ?
U 10A1. 0000.003C.0180.0800.0000.11DE  ;5117  J/T.37.SBI.ERR.NOTSET ; No, Call an error
U 10A2. 0000.003C.0180.0800.0000.11E0  ;5118  J/T.37.DO.NEXT.2 ; Yes, continue with test
            ;5119 =
```

```
            ;5120  .PAGE
            ;5121 ;
            ;5122 ;////////////////////////////////////////////////////////////////////////////////
            ;5123 ;
            ;5124 ; SBI Error register bit 13 (RDS) did not set.
            ;5125 ; (i.e., CPU did NOT detect the RDS Tag from the memory)
            ;5126 ; Call an error.
            ;5127 ;
            ;5128 =0
            ;5129 T.37.SBI.ERR.NOTSET:
U 11DE, 0000,003D,5D80,0800,0084,7054  ;5130  ERROR2[.7]
U 11DF, 0000,003C,0180,0800,0000,11E0  ;5131  NOP
            ;5132 ;
            ;5133 ;////////////////////////////////////////////////////////////////////////////////
            ;5134 ;
            ;5135 ; ERROR LOGOUT:
            ;5136 ;
            ;5137 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5138 ;  I/O Address of Configuration Register C/D
            ;5139 ;  Data Pattern Under Test
            ;5140 ;  Not Used
            ;5141 ;  Not Used
            ;5142 ;  Not Used
            ;5143 ;  ID[sbi.err] Register
            ;5144 ;
            ;5145 ;////////////////////////////////////////////////////////////////////////////////
            ;5146 ;
            ;5147  ;
            ;5148  ; Now check the Data returned from memory to see if it is correct.
            ;5149  ;
            ;5150 =0
            ;5151 T.37.DO.NEXT.2:
U 11E0, 0818,0039,0D80,0800,0000,1247  ;5152  SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC Bypass
U 11E1, 0018,0038,1980,0800,0200,137E  ;5153  VA_K[ZERO]  ; Point VA to location 0
U 137E, 0000,003C,0180,C800,0000,137F  ;5154  D[LONG]_CACHE.P  ; Read the data from the array
U 137F, 0001,003C,0180,09D8,0000,1380  ;5155  RC[0B]_D  ; Save the data in RC[0B]
            ;5156  ALU_D.XOR.RC[0C], ; Exclusive the received data with the expected
U 1380, 0011,0020,0180,0960,0010,1381  ;5157  CLK.UBCC  ; memory data.
U 1381, 0000,013C,0180,0800,0000,11E2  ;5158  Z?  ; Are they equal ?
U 11E2, 0000,003C,0180,0800,0000,11E4  ;5159 =0 J/T.37.BAD.DATA.1 ; NO, call an error
U 11E3, 0000,003C,0180,0800,0000,1382  ;5160  J/T.37.CHK.SYNDROME ; Yes, continue with test
```

```
            ;5161  .PAGE
            ;5162 ;
            ;5163 ;////////////////////////////////////////////////////////////////////////////////
            ;5164 ;
            ;5165 ; Received Data and Expected Data did not match.
            ;5166 ; (i.e., data was modified on the array for RDS error)
            ;5167 ; Call an error.
            ;5168 ;
            ;5169 =0
            ;5170 T.37.BAD.DATA.1:
U 11E4. 0000.003D.1180.0800.0084.7054  ;5171  ERROR2[.4]
U 11E5. 0000.003C.0180.0800.0000.1382  ;5172  NOP
            ;5173 ;
            ;5174 ;////////////////////////////////////////////////////////////////////////////////
            ;5175 ;
            ;5176 ; ERROR LOGOUT:
            ;5177 ;
            ;5178 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5179 ;  I/O Address of Configuration Register C/D
            ;5180 ;  Expected Data Pattern
            ;5181 ;  Received Data Pattern
            ;5182 ;
            ;5183 ;////////////////////////////////////////////////////////////////////////////////
            ;5184 ;
            ;5185  ;
            ;5186  ; Now lets check the syndrome bits and see if they are correct.  There
            ;5187  ; should be an even number of syndrome bits and if there isn't then
            ;5188  ; call an error.
            ;5189  ;
            ;5190 T.37.CHK.SYNDROME:
U 1382. 0010.0038.0180.0968.0200.1383  ;5191  VA_RC[0D]  ; Point Cnfg reg C/D to the VA
U 1383. 0000.003C.0180.CB00.0000.1384  ;5192  D[LONG]_CACHE.P  ; Read the Register
U 1384. 0001.003C.0180.09D0.0000.1385  ;5193  RC[0A]_D  ; Save it in RC[0A]
            ;5194  D_D.AND.K[.7F],  ; Mask the Syndrome bits
U 1385. 0819.0034.5980.09D8.0000.10A4  ;5195   RC[0B]_ALU ; Save them in RC[0B]
U 10A4. 0000.003D.0180.0800.0000.12CD  ;5196 =00 CALL[SYNDROME.EVEN] ; Does CNFG reg C/D have an even number of
            ;5197      ; ones in its syndrome ?
U 10A5. 0000.003C.0180.0800.0000.11E7  ;5198  J/T.37.ADJUST.RC0C ; yes, continue with test
U 10A6. 0000.003C.0180.0800.0000.11E6  ;5199  J/T.37.SYN.NOT.EVEN ; No, call an error
            ;5200
            ;5201 =
```

```
          ;5202  .PAGE
          ;5203 ;
          ;5204 ;////////////////////////////////////////////////////////////////////////////
          ;5205 ;
          ;5206 ; The syndrome bits did NOT have an even number of ones
          ;5207 ; (indicating an RDS error)
          ;5208 ; Call an error
          ;5209 ;
          ;5210 =0
          ;5211 T.37.SYN.NOT.EVEN:
U 11E6, 0000,003D,D580,0800,0084,7054  ;5212  ERROR2[.6]
          ;5213 ;
          ;5214 ;////////////////////////////////////////////////////////////////////////////
          ;5215 ;
          ;5216 ; ERROR LOGOUT:
          ;5217 ;
          ;5218 ; DATA: I/O Base Address of MS780-E Currently Under Test
          ;5219 ;  I/O Address of Configuration Register C/D
          ;5220 ;  Data Pattern Under Test
          ;5221 ;  Received Syndrome bits
          ;5222 ;  Contents of Configuration Register C/D
          ;5223 ;  Not Used
          ;5224 ;
          ;5225 ;////////////////////////////////////////////////////////////////////////////
          ;5226 ;
          ;5227  ;
          ;5228  ; Now get ready to do the next loop.  Shift the test pattern (RC[0C])
          ;5229  ; left one place with a zero shifted into bit 0.
          ;5230  ;
          ;5231 T.37.ADJUST.RC0C:
U 11E7, 0810,0038,0180,0960,0000,1386  ;5232  D_RC[0C]  ; Move Test Pattern to the D register
          ;5233  SI/ZERO,  ; Get ready to shift the test pattern left
U 1386, 0021,003C,0180,09E0,0000,1387  ;5234  RC[0C]_D.LEFT  ; one place with 0 in the low bit.
          ;5235  ;
          ;5236  ; Decrement the pattern counter and if the pattern counter is equal to
          ;5237  ; zero then Restart this test for the next controller.  If the pattern
          ;5238  ; counter is not equal to 0 then go back to the loop point and do
          ;5239  ; another pass
          ;5240  ;
U 1387, 0800,003C,0180,0A00,0000,1388  ;5241  D_R[0]  ; Load the D register with the pattern count
          ;5242  ALU_D-K[.1],  ; Decrement the pattern counter then
U 1388, 0019,0000,0580,0A80,0010,1389  ;5243  R[0]_ALU,CLK.UBCC ; restore back in R[0]
U 1389, 0000,013C,0180,0800,0000,11E8  ;5244  Z?  ; Is pattern count now equal to zero ?
U 11E8, 0000,003C,0180,0800,0000,11BC  ;5245 =0 J/T.37.LOOP.POINT ; No, continue with this pattern
U 11E9, 0000,003C,0180,0800,0000,1094  ;5246  J/RESTART.TEST.37 ; Yes, Restart this test for the next
          ;5247     ; MS780-E controller
          ;5248  ;
          ;5249  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
          ;5250  ;
          ;5251 =0
          ;5252 T.37.EXIT:
U 11EA, 0000,003D,0180,0800,0000,12B9  ;5253  CALL[MS780E.CLEANUP] ; Clear memory CNFG Regs before exitting
U 11EB, 0000,003C,0180,0800,0000,11EC  ;5254  NOP
          ;5255
```

```
;5256  .PAGE "TEST 1C3 DOUBLE BIT ERROR PART 2"
;5257 ;*************************************************************
;5258 ;+++
;5259 ;
;5260 ; Functional Description:
;5261 ; ---------------------
;5262 ;
;5263 ; This sequence of subtests further test the ability of the ECC logic to
;5264 ; detect and log RDS errors.  The errors are forced both in the check
;5265 ; bits and in the data bits as detailed below.  All of these subtests
;5266 ; make use of Diagnostic mode 2 in order to force the RDS error.
;5267 ;
;5268 ;   SUBTEST 1
;5269 ;
;5270 ; This Subtest verifies the ability of the ECC logic to detect RDS errors
;5271 ; when any of the data bits is in error.  This first subtest will float
;5272 ; two ones through a field of zeroes.
;5273 ;
;5274 ;   SUBTEST 2
;5275 ;
;5276 ; This subtest is very similar to the first one.  The difference between
;5277 ; the two is the fact that in this case two zeroes are floated through a
;5278 ; field of ones.
;5279 ;
;5280 ;   SUBTEST 3
;5281 ;
;5282 ; This subtest checks the ability of the ECC logic to detect a RDS error
;5283 ; when one error occurs in the check bits and the other in the data.
;5284 ;
;5285 ;
;5286 ; Logic Description:
;5287 ; -----------------
;5288 ; N/A
;5289 ;
;5290 ;
;5291 ; Error Logout:
;5292 ; ------------
;5293 ;
;5294 ; See individual error reports.
;5295 ;
;5296 ; Sync Point Description:
;5297 ; ----------------------
;5298 ; N/A
;5299 ;
;5300 ;=============================================================
;5301 ;
;5302 ; Register Map:
;5303 ; ------------
;5304 ;
;5305 ;     RA,RB     Description:
;5306 ;     -----     -----------
;5307 ;
;5308 ;       0       First Bit Pattern (subtests 1 and 2)
;5309 ;
;5310 ;       1       Copy of first bit pattern
```

```
            ;5311 ;
            ;5312 ;       2 Subtest Flag
            ;5313 ;
            ;5314 ;      RC       Description:
            ;5315 ;      --       -----------
            ;5316 ;
            ;5317 ;       E I/O Base Address of MS780-E Currently Under Test
            ;5318 ;
            ;5319 ;       D I/O Address of Configuration Register C/D
            ;5320 ;
            ;5321 ;       C Initialize Data Pattern
            ;5322 ;
            ;5323 ;       B Check Bit Pattern with One Bit in Error
            ;5324 ;
            ;5325 ;       A Check bit to be Complemented
            ;5326 ;
            ;5327 ;       9 Contents of Configuration Register C/D
            ;5328 ;
            ;5329 ;       1 Copy of Second Bit Pattern (subtest 1 and 2)
            ;5330 ;
            ;5331 ; 0 Second Bit Pattern (subtest 1 and 2)
            ;5332 ;
            ;5333 ;--
            ;5334 ;*************************************************************
            ;5335 =0
U 11EC, 0000,003D,0180,0800,0000,107F  ;5336 T.38: NEWTST   ; Signify the start of a new test
U 11ED, 0000,003C,0180,0800,0000,10A8  ;5337  NOP
U 10A8, 0000,003D,0180,0800,0000,10C2  ;5338 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 10A9, 0000,003C,0180,0800,0000,121E  ;5339  J/T.38.EXIT  ; No MS780-E's to test on the SBI
U 10AA, 0000,003C,0180,0800,0000,10AC  ;5340  NOP
            ;5341 =
            ;5342  ;
            ;5343  ; This is the entry point for restarting to test the next controller
            ;5344  ; on the SBI.
            ;5345  ;
            ;5346 =00
            ;5347 RESTART.TEST.38:
U 10AC, 0000,003D,0180,0800,0000,105C  ;5348  CALL[START.TEST] ; Configure the controllers
U 10AD, 0000,003C,0180,0800,0000,121E  ;5349  J/T.38.EXIT  ; No more controllers to configure/test
            ;5350 ;
            ;5351 ;=============================================================================
            ;5352 ;
            ;5353 ; Subtest 1
            ;5354 ;
U 10AE, 0003,003C,0180,0A90,0000,11EE  ;5355  R[02]_0   ; Clear Subtest flag to signify this is
            ;5356      ; subtest 1
            ;5357 =
U 11EE, 0000,003D,0180,0800,0000,1208  ;5358 =0 CALL[TEST.38.TST] ; Execute the Subtest
            ;5359 ;
            ;5360 ;=============================================================================
            ;5361 ;
            ;5362 ; Subtest 2
            ;5363 ;
U 11EF, 0018,0038,0580,0A90,0000,11F0  ;5364  R[02]_K[.1]  ; Set Subtest flag to signify this is subtest
            ;5365      ; number 2
```

```
U 11F0, 0000,003D,0180,0800,0000,1208  ;5366 =0 CALL[TEST.38.TST] ; Execute the Subtest
U 11F1, 0000,003C,0180,0800,0000,11F2  ;5367  NOP
            ;5368 ;
            ;5369 ;=============================================================================
            ;5370 ;
            ;5371 ; Subtest 3
            ;5372 ;
U 11F2, 0000,003D,0180,0800,0000,1234  ;5373 =0 SUBTEST   ; Update the subtest counter
U 11F3, 0018,0038,3180,09D0,0000,138A  ;5374  RC[0A]_K[.40]  ; Load RC[0A] with check bit to be complemented
U 138A, 0810,0038,0180,0950,0000,138B  ;5375  D_RC[0A]  ; Load D with check bit pattern
U 138B, 0019,0020,8580,09D8,0000,138C  ;5376  RC[0B]_D.XOR.K[.C] ; Generate check bit pattern with one bit in
            ;5377     ; error
U 138C, 0018,0038,0580,09E0,0000,11F4  ;5378  RC[0C]_K[.1]  ; Initialize data bit pattern
            ;5379 =0
            ;5380  ;
            ;5381  ; This is the loop point for Subtest 3 to run all test patterns through
            ;5382  ; the memory test location.
            ;5383  ;
            ;5384 T.38.S3.LOOP.POINT:
U 11F4, 0000,003D,0180,0800,0000,1234  ;5385  SUBTEST   ; Update the subtest counter
U 11F5, 0000,003C,0180,0800,0000,11F6  ;5386  NOP
U 11F6, 0000,003D,0180,0800,0000,10BC  ;5387 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11F7, 0000,003C,0180,0800,0000,11F8  ;5388  NOP
U 11F8, 0000,003D,0180,0800,0000,12B9  ;5389 =0 CALL[MS780E.CLEANUP] ; Clear the Configuration Registers
U 11F9, 0018,0038,1980,0800,0200,138D  ;5390  VA_K[ZERO]  ; Point VA to location 0
U 138D, 0810,0038,0180,0960,0000,138E  ;5391  D_RC[0C]  ; Load the D register with test pattern
U 138E, 0000,003C,0180,A800,0000,11FA  ;5392  CACHE.P_D[LONG]  ; Initialize the location 0 with test pattern
            ;5393 =0 D_RC[0B],  ; Load D with check bits that have one bit
U 11FA, 0810,0039,0180,0958,0000,12B3  ;5394  CALL[SET_ECC]  ; in error and set ECC accordingly
U 11FB, 0000,003C,0180,0800,0000,11FC  ;5395  NOP
U 11FC, 0818,0039,0980,0800,0000,1247  ;5396 =0 SET.DIAG.MODE[.2] ; Select ECC substitute mode
U 11FD, 0000,003C,0180,0800,0000,11FE  ;5397  NOP
U 11FE, 0000,003D,8580,0800,0084,726F  ;5398 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 11FF, 0018,0038,1980,0800,0200,138F  ;5399  VA_K[ZERO]  ; Point VA to location 0
U 138F, 0000,003C,0180,C800,0000,10B0  ;5400  D[LONG]_CACHE.P  ; Read the test location
U 10B0, 0000,003D,0180,0800,0000,1269  ;5401 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 10B1, 0000,003C,0180,0800,0000,1200  ;5402  J/T.38.S3.NO.UTRAP ; No, should have been, call an error
U 10B2, 0000,003C,0180,0800,0000,1201  ;5403  J/T.38.S3.CHK.BIT10 ; Yes, continue with the test
            ;5404 =
```

```
            ;5405  .PAGE
            ;5406 ;
            ;5407 ;////////////////////////////////////////////////////////////////////////////////
            ;5408 ;
            ;5409 ; Didn't get a RDS micro trap.
            ;5410 ; (i.e., memory did not detect RDS with one error in data and one
            ;5411 ; error in checkbits)
            ;5412 ; Call an error.
            ;5413 ;
            ;5414 =0
            ;5415 T.38.S3.NO.UTRAP:
U 1200, 0000,003D,0180,0800,0084,7054  ;5416  ERROR2[.8]
            ;5417 ;
            ;5418 ;////////////////////////////////////////////////////////////////////////////////
            ;5419 ;
            ;5420 ; ERROR LOGOUT:
            ;5421 ;
            ;5422 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5423 ;  I/O Address of Configuration Register C/D
            ;5424 ;  Memory Initialize Data Pattern
            ;5425 ;  Checkbit Pattern with one bit in error
            ;5426 ;  Checkbit to be Complemented
            ;5427 ;  Not Used
            ;5428 ;  Expected Micro Trap Flags
            ;5429 ;  Received Micro Trap Flags
            ;5430 ;
            ;5431 ;////////////////////////////////////////////////////////////////////////////////
            ;5432 ;
            ;5433 T.38.S3.CHK.BIT10:
U 1201, 0010,0038,0180,0968,0200,1390  ;5434  VA_RC[0D]  ; Load VA with Configuration Reg C/D address
U 1390, 0000,003C,0180,C800,0000,1391  ;5435  D[LONG]_CACHE.P  ; Read the contents of CNFG C/D and
U 1391, 0001,003C,0180,09B8,0000,1392  ;5436  RC[7]_D  ; ...save in RC[09]
U 1392, 0000,003C,F580,0800,0084,7393  ;5437  SC_K[.A]  ; Get read to mask out bit 10 ( RDS Flag )
U 1393, 0003,001C,0180,09C8,0000,1394  ;5438  RC[9]_NOT.MASK  ; Bit expected to be set in CNFG Reg C/D
            ;5439  D_D.ANDNOT.MASK, ; Mask out Bit 10 from Configuration Reg C/D
U 1394, 0801,0024,0180,0800,0010,1395  ;5440  CLK.UBCC  ; for the zero test
U 1395, 0001,013C,0180,09C0,0000,1202  ;5441  Z?,RC[8]_D  ; Was bit 10 Set ?
U 1202, 0000,003C,0180,0800,0000,1205  ;5442 =0 J/T.38.S3.BIT10.SET ; Yes, continue with the test
U 1203, 0000,003C,0180,0800,0000,1204  ;5443  NOP  ; No, Should have been, call an error
```

```
            ;5444  .PAGE
            ;5445 ;
            ;5446 ;////////////////////////////////////////////////////////////////////////////
            ;5447 ;
            ;5448 ; Bit 10 ( RDS Flag ) in Configuration Register C/D was not set.  Call
            ;5449 ; an error.
            ;5450 ;
            ;5451 =0
U 1204, 0000,003D,D580,0800,0084,7054  ;5452  ERROR2[.6]
            ;5453 ;
            ;5454 ;////////////////////////////////////////////////////////////////////////////
            ;5455 ;
            ;5456 ; ERROR LOGOUT:
            ;5457 ;
            ;5458 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5459 ;  I/O Address of Configuration Register C/D
            ;5460 ;  Initialize Data Pattern
            ;5461 ;  Checkbit Pattern with one bit in error
            ;5462 ;  Checkbit to be Complemented
            ;5463 ;  Expected bit to be set in CNFG Reg C/D
            ;5464 ;  Received Status of the bit in CNFG Reg C/D
            ;5465 ;  Contents of Configuration Register C/D
            ;5466 ;
            ;5467 ;////////////////////////////////////////////////////////////////////////////
            ;5468 ;
            ;5469 T.38.S3.BIT10.SET:
U 1205, 0010,0038,0180,0960,0010,1396  ;5470  ALU_RC[0C],CLK.UBCC ; Check and see where the floating one is
            ;5471     ; ...in the data pattern
U 1396, 0000,1B3C,0180,0800,0000,1047  ;5472  ALU.N?   ; Is the pattern negative ?
U 1047, 0000,003C,0180,0800,0000,139B  ;5473 =0111 J/T.38.S3.NOT.NEGATIVE ; No, go shift data pattern left w/zeros
U 104F, 0010,0038,01C0,0950,0000,1397  ;5474  Q_RC[0A]  ; Yes, get ready to adjust check bits & pattern
            ;5475  SI/ZERO,  ; Set up to shift in zeroes and then shift
U 1397, 0041,203C,0180,09D0,0000,1398  ;5476   RC[0A]_Q.RIGHT ; ...the check bit pattern to be complemented
U 1398, 0810,0038,0180,0950,0000,1399  ;5477  D_RC[0A]  ; Load D with check bit pattern
U 1399, 0019,0020,8580,09D8,0000,139A  ;5478  RC[0B]_D.XOR.K[.C] ; Generate check bit pattern with one bit in
            ;5479     ; error
            ;5480  RC[0C]_K[.1],  ; Re-initialize the data pattern and then
U 139A, 0018,0038,0580,09E0,0000,139D  ;5481  J/T.38.S3.CHK.RC0A ; Go and check RC[0A] to see if equal to 0
            ;5482 T.38.S3.NOT.NEGATIVE:
U 139B, 0810,0038,0180,0960,0000,139C  ;5483  D_RC[0C]  ; Load D with the test data pattern
            ;5484  SI/ZERO,  ; Set up to shift zeroes and then shift
U 139C, 0021,003C,0180,09E0,0000,139D  ;5485  RC[0C]_D.LEFT  ; the data pattern left one place w/zeroes
            ;5486 T.38.S3.CHK.RC0A:
U 139D, 0010,0038,0180,0950,0010,139E  ;5487  ALU_RC[0A],CLK.UBCC ; Move the check bit pattern to the ALU
U 139E, 0000,013C,0180,0800,0000,1206  ;5488  Z?   ; Is the check bit pattern = 0 ?
U 1206, 0000,003C,0180,0800,0000,11F4  ;5489 =0 J/T.38.S3.LOOP.POINT ; No, go execute another pass
U 1207, 0000,003C,0180,0800,0000,10AC  ;5490  J/RESTART.TEST.38 ; Yes, restart this test for next controller
            ;5491  ;
            ;5492  ; Subtests one and two of this test will be run as a subroutine.  The
            ;5493  ; following code is that subroutine.
            ;5494  ;
            ;5495 =0
            ;5496 TEST.38.TST:
U 1208, 0000,003D,0180,0800,0000,1234  ;5497  SUBTEST   ; Update the Subtest counter
U 1209, 0018,0038,0580,0A80,0000,139F  ;5498  R[0]_K[.1]  ; Initialize first bit pattern
```

```
U 139F. 0018.0038.0980.0980.0000.13A0  ;5499  RC[0]_K[.2]  ; Initialize second bit pattern
            ;5500  ;
            ;5501  ; This is the loop point for subtests 1 and 2 to run all necessary
            ;5502  ; test patterns through the memory test location.
            ;5503  ;
            ;5504 T.38.SX.LOOP.POINT:
U 13A0. 0000.003C.0180.0A10.0010.13A1  ;5505  ALU_R[2],CLK.UBCC ; Move the subtest flag to the ALU
U 13A1. 0000.013C.0180.0800.0000.120A  ;5506  Z?   ; Is the subtest flag equal to 0 ?
U 120A. 0000.003C.0180.0800.0000.13A7  ;5507 =0 J/T.38.DO.SUBTEST.2 ; No, that means this is subtest 2
            ;5508  ;
            ;5509  ; Doing Subtest 1.  Set-up the data necessary to execute subtest 1.
            ;5510  ;
U 120B. 0810.0038.0180.0900.0000.13A2  ;5511  D_RC[0]   ; yes, this is subtest 2, move pattern 1 to D
U 13A2. 0001.003C.0180.0988.0000.13A3  ;5512  RC[01]_D  ; Load RC[01] with pattern 2
U 13A3. 0800.003C.0180.0A00.0000.13A4  ;5513  D_R[0]   ; move pattern 1 to the D register
U 13A4. 0001.003C.0180.0A88.0000.13A5  ;5514  R[1]_D   ; Load R[1] with pattern 1
U 13A5. 0010.0038.01C0.0908.0000.13A6  ;5515  Q_RC[01]  ; Move pattern 2 to the Q register
            ;5516  RC[0C]_D.OR.Q,  ; OR the patterns together and
U 13A6. 001D.0030.0180.09E0.0000.120C  ;5517  J/T.38.SX.DO.SUBTEST ; then execute the subtest code
            ;5518  ;
            ;5519  ; Doing Subtest 2.  Set-up the data necessary to execute subtest 2.
            ;5520  ;
            ;5521 T.38.DO.SUBTEST.2:
U 13A7. 0810.0038.0180.0900.0000.13A8  ;5522  D_RC[0]   ; Move pattern 2 to the D register
U 13A8. 0001.0028.0180.0988.0000.13A9  ;5523  RC[01]_NOT.D  ; Load RC[01] with the complement of pattern 1
U 13A9. 0800.003C.0180.0A00.0000.13AA  ;5524  D_R[0]   ; Move pattern 1
U 13AA. 0001.0028.0180.0A88.0000.13AB  ;5525  R[1]_NOT.D  ; Load R[1] with the complement of pattern 2
U 13AB. 0800.003C.0180.0A08.0000.13AC  ;5526  D_R[1]   ; Load D with complemented pattern 2
U 13AC. 0010.0038.01C0.0908.0000.13AD  ;5527  Q_RC[1]   ; Load Q with complemented pattern 1
U 13AD. 001D.0034.0180.09E0.0000.120C  ;5528  RC[0C]_D.AND.Q  ; AND the patterns together and execute subtest
            ;5529 =0
            ;5530 T.38.SX.DO.SUBTEST:
U 120C. 0000.003D.0180.0800.0000.122C  ;5531  ERLOOP   ; Loop on error from here
U 120D. 0000.003C.0180.0800.0000.120E  ;5532  NOP
U 120E. 0000.003D.0180.0800.0000.10BC  ;5533 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 120F. 0000.003C.0180.0800.0000.1210  ;5534  NOP
U 1210. 0000.003D.0180.0800.0000.12B9  ;5535 =0 CALL[MS780E.CLEANUP] ; Clean up the Configuration Registers
U 1211. 0018.0038.1980.0800.0200.13AE  ;5536  VA_K[ZERO]  ; Point VA to location 0
U 13AE. 0810.0038.0180.0960.0000.13AF  ;5537  D_RC[0C]  ; Move data pattern to the D reg for write
U 13AF. 0000.003C.0180.A800.0000.1212  ;5538  CACHE.P_D[LONG]  ; Write the data pattern to location 0
U 1212. 0818.0039.8580.0800.0000.12B3  ;5539 =0 SET.ECC[.C]  ; Set substitute ECC bits to xC
U 1213. 0000.003C.0180.0800.0000.1214  ;5540  NOP
U 1214. 0818.0039.0980.0800.0000.1247  ;5541 =0 SET.DIAG.MODE[.2] ; Select diagnostic mode ECC substitute mode
U 1215. 0000.003C.0180.0800.0000.1216  ;5542  NOP
U 1216. 0000.003D.8580.0800.0084.726F  ;5543 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 1217. 0018.0038.1980.0800.0200.13B0  ;5544  VA_K[ZERO]  ; Point VA to location 0
U 13B0. 0000.003C.0180.C800.0000.10B4  ;5545  D[LONG]_CACHE.P  ; Read the location to force the error
U 10B4. 0000.003D.0180.0800.0000.1269  ;5546 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 10B5. 0000.003C.0180.0800.0000.1218  ;5547  J/T.38.SX.NO.UTRAP ; No, should have been, call an error
U 10B6. 0000.003C.0180.0800.0000.1219  ;5548  J/T.38.SX.CHK.BIT10 ; Yes, continue with test
            ;5549 =
```

```
            ;5550  .PAGE
            ;5551 ;
            ;5552 ;////////////////////////////////////////////////////////////////////////////////
            ;5553 ;
            ;5554 ; Didn't get a RDS micro trap on the read that forced the error.
            ;5555 ; Call an error.
            ;5556 ;
            ;5557 =0
            ;5558 T.38.SX.NO.UTRAP:
U 1218, 0000,003D,0180,0800,0084,7054  ;5559  ERROR2[.8]
            ;5560 ;
            ;5561 ;////////////////////////////////////////////////////////////////////////////////
            ;5562 ;
            ;5563 ; ERROR LOGOUT:
            ;5564 ;
            ;5565 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5566 ;  I/O Address of Configuration Register C/D
            ;5567 ;  Memory Initialize Data Pattern
            ;5568 ;  Not Used
            ;5569 ;  Not Used
            ;5570 ;  Not Used
            ;5571 ;  Expected Micro Trap Flags
            ;5572 ;  Received Micro Trap Flags
            ;5573 ;
            ;5574 ;////////////////////////////////////////////////////////////////////////////////
            ;5575 ;
            ;5576 T.38.SX.CHK.BIT10:
U 1219, 0010,0038,0180,0968,0200,13B1  ;5577  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 13B1, 0000,003C,0180,C800,0000,13B2  ;5578  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 13B2, 0001,003C,0180,09C8,0000,13B3  ;5579  RC[09]_D  ; save in RC[09]
U 13B3, 0000,003C,F580,0800,0084,73B4  ;5580  SC_K[.A]  ; Get read to mask out bit 10 ( RDS Flag )
U 13B4, 0003,001C,0180,09D8,0000,13B5  ;5581  RC[0B]_NOT.MASK  ; Expected bit to be set in CNFG Reg C/D
            ;5582  D_D.ANDNOT.MASK, ; Mask out Bit 10 from Configuration Reg C/D
U 13B5, 0801,0024,0180,0800,0010,13B6  ;5583  CLK.UBCC ; ...for the zero test
U 13B6, 0001,013C,0180,09D0,0000,121A  ;5584  Z?,RC[0A]_D  ; Was bit 10 Set ?
U 121A, 0000,003C,0180,0800,0000,121D  ;5585 =0 J/T.38.SX.BIT10.SET ; Yes, continue with the test
U 121B, 0000,003C,0180,0800,0000,121C  ;5586  NOP  ; No, Should have been, call an error
```

```
            ;5587  .PAGE
            ;5588 ;
            ;5589 ;////////////////////////////////////////////////////////////////////////////
            ;5590 ;
            ;5591 ; Bit 10 ( RDS Flag ) in Configuration Register C/D was not set.  Call
            ;5592 ; an error.
            ;5593 ;
            ;5594 =0
U 121C. 0000,003D,D580,0800,0084,7054  ;5595  ERROR2[.6]
            ;5596 ;
            ;5597 ;////////////////////////////////////////////////////////////////////////////
            ;5598 ;
            ;5599 ; ERROR LOGOUT:
            ;5600 ;
            ;5601 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5602 ;  I/O Address of Configuration Register C/D
            ;5603 ;  Memory Initialize Data Pattern
            ;5604 ;  Expected bit to be set in CNFG Reg C/D
            ;5605 ;  Received Status of the bit in CNFG Reg C/D
            ;5606 ;  Contents of Configuration Register C/D
            ;5607 ;
            ;5608 ;////////////////////////////////////////////////////////////////////////////
            ;5609 ;
            ;5610  ;
            ;5611  ; Now check the test data patterns and see if we need to return from
            ;5612  ; here or to continue with this subtest
            ;5613  ;
            ;5614 T.38.SX.BIT10.SET:
U 121D. 0010,0038,0180,0900,0010,13B7  ;5615  ALU_RC[0],CLK.UBCC ; Move data pattern to the ALU and check
U 13B7. 0000,1B3C,0180,0800,0000,1057  ;5616  ALU.N?   ; to see if it is negative
U 1057. 0000,003C,0180,0800,0000,13BC  ;5617 =0111 J/T.38.SX.NOT.NEGATIVE ; No, shift left RC[0] with zeroes
U 105F. 0800,003C,0180,0A00,0000,13B8  ;5618  D_R[0]   ; Yes, shift left R[0] w/zeroes
            ;5619  SI/ZERO,  ; prepare to shift in zeroes
U 13B8. 0500,003C,0180,0800,0000,13B9  ;5620    D_D.LEFT  ; and then do the shift
U 13B9. 0001,003C,0180,0A80,0000,13BA  ;5621  R[0]_D   ; Restore in R[0]
            ;5622  D_D.LEFT,  ; Shift the D one more to cause the RDS
U 13BA. 0500,003C,0180,0800,0000,13BB  ;5623   SI/ZERO  ; ...shifting in zeroes
            ;5624  RC[00]_D,  ; Save in RC[0] then go check R[0]
U 13BB. 0001,003C,0180,0980,0000,13BF  ;5625   J/T.38.SX.CHK.R0 ; ...
            ;5626 T.38.SX.NOT.NEGATIVE:
U 13BC. 0810,0038,0180,0900,0000,13BD  ;5627  D_RC[0]   ; Load data pattern in D
            ;5628          SI/ZERO,  ; Get ready to shift in zeroes
U 13BD. 0500,003C,0180,0800,0000,13BE  ;5629    D_D.LEFT  ; and shift left the data pattern w/zeroes
U 13BE. 0001,003C,0180,0980,0000,13BF  ;5630  RC[0]_D   ; Restore in RC[0]
            ;5631 T.38.SX.CHK.R0:
U 13BF. 0000,003C,0180,0A00,0010,13C0  ;5632  ALU_R[0],CLK.UBCC ; Move first data pattern to the ALU
U 13C0. 0000,1B3C,0180,0800,0000,1067  ;5633  ALU.N?   ; Is the data pattern NEGATIVE ?
U 1067. 0000,003C,0180,0800,0000,13A0  ;5634 =0111 J/T.38.SX.LOOP.POINT ; No, continue with next loop
U 106F. 0000,003E,0180,0800,0000,0001  ;5635  RETURN1
            ;5636  ;
            ;5637  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;5638  ;
            ;5639 =0
            ;5640 T.38.EXIT:
U 121E. 0000,003D,0180,0800,0000,12B9  ;5641  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
```

```
U 121F, 0000,003C,0180,0800,0000,1220  ;5642  NOP
            ;5643
            ;5644
```

```
            ;5645  .PAGE "TEST 1C4 RESERVED"
            ;5646 =0
U 1220. 0000.003D.0180.0800.0000.107F  ;5647 T1C4: NEWTST
U 1221. 0000.003C.0180.0800.0000.1222  ;5648  NOP
            ;5649
```

```
            ;5650  .PAGE "TEST 1C5 RESERVED"
            ;5651 =0
U 1222. 0000.003D.0180.0800.0000.107F  ;5652 T1C5: NEWTST
U 1223. 0000.003C.0180.0800.0000.1224  ;5653  NOP
            ;5654
```

```
            ;5655  .PAGE "TEST 1C6 RESERVED"
            ;5656 =0
U 1224, 0000,003D,0180,0800,0000,107F  ;5657 T1C6: NEWTST
U 1225, 0000,003C,0180,0800,0000,13C1  ;5658  NOP
            ;5659
```

```
            ;5660   .PAGE "DUMMY NEWTST"
U 13C1, 0000,003D,0180,0800,0000,107F   ;5661 DUM: NEWTST
```

: Location / Line Number Index

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2755= 2756= 2757= 2760=
U 1008  2356= 2358= 2360= 2365= 2345= 2346= 2347= 2348=
U 1010  2379= 2381= 2382= 2386= 2402= 2403= 2404= 2412=
U 1018  2882= 2883= 2884= 2885= 2886= 2887= 2888= 2889=
U 1020  2414= 2416= 2418= 2419= 2462= 2463= 2464= 2466=
U 1028  2510= 2511= 2512= 2514= 1912= 1913= 3240= 3241=
U 1030  3329= 3330= 3331= 1902 3340= 3341= 3343= 3344=
U 1038  3526= 3527= 3528= 1903 1965= 1967= 3247= 3248=
U 1040  3537= 3538= 3539= 3541= 1978= 1979= 1904 5473=
U 1048  3604= 3605= 3606= 1906 2039= 2040= 2021: 5474=
U 1050  3686= 3688= 3690= 1907 2085= 2086= 1914 5617=
U 1058  2208= 2209= 2067: 1915 2334= 2335= 2101: 5618=
U 1060  3898= 3899= 3900= 1916 2337= 2342= 1917 5634=
U 1068  3908= 3909= 3910= 3917= 2572= 2577= 1918 5635=
U 1070  3969= 3970= 3971= 1919 4197= 4198= 4199= 1920
U 1078  4210= 4211= 4212= 4218= 4291= 4292= 4293= 1954
U 1080  4520= 4521= 4522= 1955 4532= 4533= 4534= 1956
U 1088  4565= 4566= 4567= 1957 4685= 4686= 4687= 1958
U 1090  4904= 4905= 4906= 1959 4917= 4918= 4919= 4921=
U 1098  4950= 4951= 4952= 1960 5079= 5080= 5081= 1961
U 10A0  5116= 5117= 5118= 1962 5196= 5198= 5199= 1963
U 10A8  5338= 5339= 5340= 1964 5348= 5349= 5355= 1968
U 10B0  5401= 5402= 5403= 1969 5546= 5547= 5548= 1970
U 10B8  2582= 2587= 2592= 2593= 2637= 2638= 2715= 2716=
U 10C0  2794= 2795= 2861= 2862= 2870= 2871= 2872= 2874=
U 10C8  2877= 2878= 2919= 2920= 2921= 2922= 2929= 2930=
U 10D0  2936= 2938= 2941= 2942= 2961= 2962= 2964= 2965=
U 10D8  3054= 3055= 3058= 3059= 3070= 3071= 3074= 3075=
U 10E0  3205= 3206= 3245= 3246= 3327= 3328= 3359= 3360=
U 10E8  3365= 3366= 3371= 3372= 3373= 3374= 3392= 3393=
U 10F0  3404= 3428= 3435= 3436= 3443= 3444= 3524= 3525=
U 10F8  3551= 3557= 3563= 3564= 3570= 3571= 3578= 3579=
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895: 1971 3584= 3585= 1896: 1897: 1898: 1899:
U 1110  3586= 3588= 3591= 3592= 3593= 3594= 3600= 3601=
U 1118  3617= 3640= 3648= 3649= 3658= 3680= 3682= 3683=
U 1120  3701= 3702= 3725= 3726= 3732= 3733= 3734= 3735=
U 1128  3736= 3738= 3741= 3742= 3743= 3744= 3756= 3757=
U 1130  3766= 3787= 3789= 3793= 3816= 3817= 3823= 3824=
U 1138  3896= 3897= 3919= 3925= 3927= 3928= 3935= 3936=
U 1140  3950= 3951= 3952= 3953= 3954= 3955= 3956= 3957=
U 1148  3961= 3962= 3982= 4004= 4008= 4009= 4020= 4021=
U 1150  4042= 4043= 4049= 4050= 4061= 4077= 4079= 4083=
U 1158  4106= 4107= 4113= 4114= 4195= 4196= 4222= 4228=
U 1160  4232= 4233= 4240= 4241= 4254= 4255= 4260= 4261=
U 1168  4269= 4270= 4272= 4274= 4279= 4280= 4281= 4282=
U 1170  4283= 4284= 4304= 4326= 4330= 4331= 4342= 4343=
U 1178  4364= 4365= 4371= 4372= 4383= 4384= 4407= 4411=
U 1180  4434= 4435= 4441= 4442= 4518= 4519= 4542= 4543=
U 1188  4544= 4545= 4546= 4547= 4548= 4549= 4556= 4557=
U 1190  4561= 4562= 4578= 4596= 4604= 4605= 4614= 4633=
U 1198  4640= 4641= 4650= 4666= 4668= 4669= 4670= 4671=
```

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

```
U 11A0   4676= 4677= 4678= 4680= 4698= 4716= 4720= 4721=
U 11A8   4732= 4733= 4750= 4751= 4757= 4758= 4768= 4769=
U 11B0   4786= 4787= 4797= 4798= 4809= 4825= 4832= 4833=
U 11B8   4839= 4840= 4902= 4903= 4931= 4932= 4933= 4934=
U 11C0   4935= 4936= 4937= 4938= 4939= 4940= 4943= 4944=
U 11C8   4964= 4986= 4994= 4995= 5005= 5025= 5031= 5032=
U 11D0   5041= 5042= 5063= 5064= 5065= 5066= 5067= 5068=
U 11D8   5069= 5070= 5073= 5074= 5092= 5093= 5130= 5131=
U 11E0   5152= 5153= 5159= 5160= 5171= 5172= 5212= 5232=
U 11E8   5245= 5246= 5253= 5254= 5336= 5337= 5358= 5364=
U 11F0   5366= 5367= 5373= 5374= 5385= 5386= 5387= 5388=
U 11F8   5389= 5390= 5394= 5395= 5396= 5397= 5398= 5399=
U 1200   5416= 5434= 5442= 5443= 5452= 5470= 5489= 5490=
U 1208   5497= 5498= 5507= 5511= 5531= 5532= 5533= 5534=
U 1210   5535= 5536= 5539= 5540= 5541= 5542= 5543= 5544=
U 1218   5559= 5577= 5585= 5586= 5595= 5615= 5641= 5642=
U 1220   5647= 5648= 5652= 5653= 5657= 5658= 1972 1973
U 1228   1974 1975 1976 1977 1996 2019 2020 2065
U 1230   2066 2149 2150 2151 2169 2171 2197 2198
U 1238   2199 2200 2201 2202 2203 2204 2206 2207
U 1240   2343 2344 2369 2389 2390 2391 2398 2538
U 1248   2539 2540 2541 2542 2543 2567 2615 2639
U 1250   2640 2641 2642 2660 2661 2662 2663 2681
U 1258   2682 2683 2684 2710 2744 2746 2747 2748
U 1260   2750 2751 2752 2753 2754 2761 2762 2763
U 1268   2764 2787 2788 2789 2790 2792 2793 2822
U 1270   2823 2824 2863 2864 2865 2866 2875 2876
U 1278   2880 2881 2894 2895 2897 2898 2899 2901
U 1280   2906 2907 2908 2910 2915 2917 2918 2926
U 1288   2927 2931 2932 2933 2934 2935 2943 2944
U 1290   2945 2947 2948 2949 2950 2951 2952 2957
U 1298   2959 2960 2963 2988 2989 2990 2991 3019
U 12A0   3020 3022 3023 3024 3050 3052 3053 3057
U 12A8   3061 3062 2127: 3063 3064 3066 3067 3068
U 12B0   3069 3072 3073 3099 3100 3101 3102 3103
U 12B8   3104 3151 3153 3154 3155 3157 3159 3160
U 12C0   3162 3164 3165 3166 3167 3170 3171 3172
U 12C8   3175 3200 3201 3203 3204 3235 3236 3238
U 12D0   3243 3244 3345 3346 3347 3361 3362 3363
U 12D8   3364 3375 3376 3382 3384 3385 3386 3388
U 12E0   3391 3430 3431 3433 3434 3547 3548 3550
U 12E8   3558 3559 3561 3562 3589 3590 3602 3603
U 12F0   3641 3642 3643 3645 3647 3684 3727 3729
U 12F8   3730 3731 3739 3740 3745 3746 3747 3748
U 1300   3749 3751 3752 3754 3755 3788 3797 3802
U 1308   3804 3811 3814 3815 3918 3926 3937 3938
U 1310   3939 3940 3947 3948 3949 3963 3964 4006
U 1318   4007 4044 4045 4047 4048 4078 4087 4092
U 1320   4094 4101 4104 4105 4219 4220 4229 4230
U 1328   4242 4243 4244 4245 4251 4252 4253 4263
U 1330   4266 4267 4268 4285 4286 4328 4329 4366
U 1338   4367 4369 4370 4405 4406 4415 4420 4422
U 1340   4429 4432 4433 4536 4551 4552 4553 4554
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348   4555 4558 4559 4560 4563 4564 4597 4598
U 1350   4599 4600 4602 4603 4635 4636 4638 4639
U 1358   4673 4674 4681 4682 4683 4684 4718 4719
U 1360   4752 4753 4755 4756 4788 4789 4790 4791
U 1368   4792 4793 4795 4796 4826 4827 4829 4831
U 1370   4941 4942 4987 4988 4989 4990 4992 4993
U 1378   5026 5027 5029 5030 5071 5072 5154 5155
U 1380   5157 5158 5191 5192 5193 5195 5234 5241
U 1388   5243 5244 5375 5376 5378 5391 5392 5400
U 1390   5435 5436 5437 5438 5440 5441 5472 5476
U 1398   5477 5478 5481 5483 5485 5487 5488 5499
U 13A0   5505 5506 5512 5513 5514 5515 5517 5522
U 13A8   5523 5524 5525 5526 5527 5528 5537 5538
U 13B0   5545 5578 5579 5580 5581 5583 5584 5616
U 13B8   5620 5621 5623 5625 5627 5629 5630 5632
U 13C0   5633 5661
U 13C8 - 13EF Unused
U 13F0   2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8   2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR4B	978
Used	978
Remaining	

Total microwords used in memory U: 978
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4B.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 1356
ESKAR4C.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1850 Micro Trap Handler and LSI-11 Support Routines
; 1851   Micro Trap Handler
; 1931 Micro Monitor Interface Routines
; 1932   Newtest Routine
; 1988   Error Loop Routine
; 2005   Error 1 Routine
; 2030   ERROR1 WITH PARAMETER
; 2051   Error 2 Routine
; 2077   ERROR 2 WITH PARAMETER
; 2096   Micro-Monitor Call
; 2110   Reserved Control Store
; 2137   Set Argument Count
; 2160   Increment Subtest Number
; 2180 Diagnostic Specific Subroutines
; 2181   Select Controller
; 2219   START TEST
; 2428   SELECT LOWER CONTROLLER
; 2476   SELECT UPPER CONTROLLER
; 2524   Set Diagnostic Mode Routine
; 2553   SBI Unjam Routine
; 2602   RESET SUBTEST NUMBER
; 2624   Initialize the SBI
; 2651   Disable Cache Routine
; 2672   Enable Cache
; 2693   Wait Loop Routine
; 2726   Check for Interrupt Routine
; 2773   CHECK UTRAP
; 2805   Set Trap Mask
; 2833   FIND MS780-E MEMORY
; 2975   Generate IO Address
; 3000   Set Starting Address
; 3034   ENABLE NEXT MS780-E
; 3085   Set ECC Bits
; 3115   MS780-E/H CNFG REG CLEANUP
; 3185   CHECK SBI.ERR
; 3216   64K OR 256K CHIPS
; 3271   GET MEMORY SIZE
; 3330   FORCE CRD-RDS ERROR AND READ
; 3397 TEST 1C7 CRD INHIBIT
; 3668 TEST 1C8 ERROR LOG AND HIGH ERROR RATE BITS TEST
; 4053 TEST 1C9 ERROR LOGGING (CRD/RDS) TEST
; 4388 TEST 1CA ERROR ADDRESS REGISTER TEST
; 4715 TEST 1CB SINGLE/DOUBLE BIT ERROR DETECT. ON EXT READS
; 5545 TEST 1CC RESERVED
; 5550 TEST 1CD RESERVED
; 5555 TEST 1CE RESERVED
; 5560 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR4C
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ; MODIFIED BY:
;1836  ;
;1837  ;  1.1 March 10, 1983  Paul Hakala
;1838  ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839  ;   the correct memory address when 256k ram chips are
;1840  ;   present.
;1841  ;
;1842  ;
;1843  ;
;1844  ;
;1845  ;
;1846  ;*****************************************************************************
;1847  ;
```

```
;1848  .PAGE
;1849
;1850  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1851  .TOC "   Micro Trap Handler"
;1852 ;++
;1853 ; FUNCTIONAL DESCRIPTION:
;1854 ;
;1855 ; This routine is responsible for 'catching' micro-traps
;1856 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1857 ; contains the Trap-Expected Mask. If the specified bit is set in
;1858 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1859 ; an error call is made.
;1860 ;
;1861 ; INPUT PARAMETERS:
;1862 ;
;1863 ; R[0F] - Expected Trap Mask
;1864 ;  Trap Code Function
;1865 ;  ---------     --------
;1866 ;       0  System Init
;1867 ;       1  Data Unaligned
;1868 ;       2  Cross Page
;1869 ;       3  TB Modify
;1870 ;       4  TB Protection
;1871 ;       6  TB Miss
;1872 ;       7  TB Parity
;1873 ;       8  Cache Parity
;1874 ;       9  Reserved Floating Operand
;1875 ;       C  Read Data Substitute
;1876 ;       D  Time out
;1877 ;       F  Control Store Parity Error
;1878 ;      10  Odd Address
;1879 ;
;1880 ; OUTPUT PARAMETERS:
;1881 ;
;1882 ; R[0F] - Mask bit is cleared.
;1883 ;
;1884 ; DATA: Micro PC of Micro Trap
;1885 ;  Trap Code (See Above)
;1886 ;  Fault Status Register
;1887 ;  SBI Error Register
;1888 ;  Timeout Address Register
;1889 ;  Maintenance Register
;1890 ;  Cache Parity Register
;1891 ;  TB Error Register 1
;1892 ;  TB Error Register 0
;1893 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1894 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1895 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1896 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1897 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1898 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1899 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1900 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1901 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1902 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C. 0000.003C.8580.0800.0084.7001  ;1903 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D. 0000.003C.8980.0800.0084.7001  ;1904 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E. 0000.003C.6580.0800.0084.7001  ;1905 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F. 0000.003C.6180.0800.0084.7003  ;1906 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1907
U 1001. 0000.003C.81F0.2C00.0000.1037  ;1908 trph0: q_id[ustack]  ; Get upc of trap
U 1037. 0C00.003C.01E0.0800.0000.1043  ;1909  q_d,d_q   ; Setup to put it back on stack
U 1043. 0C00.003C.81E0.3C00.0000.104B  ;1910  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 104B. 0000.003C.01C0.0A78.0000.104F  ;1911  q_r[0f]   ; Get the Expected Trap Mask
          ;1912  alu_q.andnot.mask,
U 104F. 0001.2024.0180.0800.0010.1057  ;1913   clk.ubcc ; Was trap expected?
U 1057. 0000.013C.0180.0800.0000.1002  ;1914  z?
          ;1915 =0 r[0f]_alu.  ; It was, clear the mask and return
          ;1916      alu_q.and.mask, ; ...
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1917       return0  ; ...
U 1003. 0018.0038.1D80.09E8.0000.104C  ;1918 trph1: rc[0d]_sc  ; Output the Trap Code
U 104C. 0000.003D.D980.0800.0084.722D  ;1919 =0 setrcf[.9]  ; Set output count
U 104D. 0000.003C.49F0.2C00.0000.105B  ;1920  q_id[tber0]  ; Output all the error registers
U 105B. 0001.203C.4DF0.2DB0.0000.105F  ;1921  rc[6]_q,q_id[tber1] ; ...
U 105F. 0001.203C.65F0.2DB8.0000.1063  ;1922  rc[7]_q,q_id[sbi.err] ; ...
U 1063. 0001.203C.6DF0.2DD8.0000.106B  ;1923  rc[0b]_q,q_id[fault] ; ...
U 106B. 0001.203C.75F0.2DE0.0000.1077  ;1924  rc[0c]_q,q_id[maint] ; ...
U 1077. 0001.203C.79F0.2DC8.0000.1083  ;1925  rc[9]_q,q_id[parity] ; ...
U 1083. 0001.203C.69F0.2DC0.0000.108F  ;1926  rc[8]_q,q_id[time.addr] ; ...
U 108F. 0001.203C.81F0.2DD0.0000.1000  ;1927  rc[0a]_q,q_id[ustack] ; ...
          ;1928 1000: ERROR1[.C],  ;
U 1000. 0001.203D.8580.09F0.0084.709C  ;1929   rc[0e]_q ; Report the error
```

```
                ;1930  .PAGE
                ;1931  .TOC " Micro Monitor Interface Routines"
                ;1932  .TOC "   Newtest Routine"
                ;1933 ;++
                ;1934 ; FUNCTIONAL DESCRIPTION:
                ;1935 ;
                ;1936 ; This routine Initializes the following registers:
                ;1937 ;  PSL    to 1F
                ;1938 ;  CES    to 0
                ;1939 ;  ACC.CS   to disable accelleretor
                ;1940 ;  SIR    to 0
                ;1941 ;  CLK.CS   to stop interval timer
                ;1942 ;  TBER0   to disable the TB
                ;1943 ;  SBI.MAINT to    0
                ;1944 ;  SBI.ERR   to 0
                ;1945 ;  FAULT   to 0
                ;1946 ;  COMP    to 0
                ;1947 ;  RC[0E]    to  0
                ;1948 ;  R[0F]    to 0
                ;1949 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1950 ; A SCOPE call is then made to the Monitor.
                ;1951 ;
                ;1952 ; CALLING CONSTRAINT:
                ;1953 ;
                ;1954 ; =0
                ;1955 ;
                ;1956 ; SIDE EFFECTS:
                ;1957 ;
                ;1958 ; D Reg is destroyed
                ;1959 ; SC is destroyed
                ;1960 ;--
U 1093, 0000,003C,65F8,0800,0084,7097   ;1961 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1097, 0818,0038,8D80,0800,0000,1109   ;1962  d_k[.1f]  ; D=1F
U 1109, 0D00,003C,0180,0800,0000,1216   ;1963  d_dal.sc  ; D=001F0000
U 1216, 0000,003C,3D80,3C00,0000,1217   ;1964  id[psl]_d  ; psl = 001F0000
U 1217, 0818,0038,0580,0800,0000,1218   ;1965  d_k[.1]   ; Clear the CES register
U 1218, 0D00,003C,0180,0800,0000,1219   ;1966  d_dal.sc  ; D = 00010000
U 1219, 0F00,003C,3180,3C00,0000,121A   ;1967  id[ces]_d,d_0  ; ces = 00010000
U 121A, 0000,003C,5D80,3C00,0000,121B   ;1968  id[acc.cs]_d  ; acc.cs = 0
U 121B, 0000,003C,3980,3C00,0000,121C   ;1969  id[sir]_d  ; sir = 0
U 121C, 0000,003C,2980,3C00,0000,121D   ;1970  id[clk.cs]_d  ; clk.cs = 0
U 121D, 0000,003C,4980,3C00,0000,1058   ;1971  id[tber0]_d  ; tber0 = 0
U 1058, 0000,003D,0180,0800,0000,1249   ;1972 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1973  intrpt.strobe,  ; Strobe interrupts
U 1059, 0818,0038,C180,0800,4000,121E   ;1974       d_k[.ffff]  ; Setup to clear SBI error register and
U 121E, 0801,0028,6580,3C00,0000,121F   ;1975  id[sbi.err]_d,d_not.d ; and fault register
U 121F, 0F00,003C,6D80,3C00,0000,1220   ;1976  id[fault]_d,d_0  ; ...
U 1220, 0000,003C,6D80,3C00,0000,1221   ;1977  id[fault]_d  ; fault = 0
U 1221, 0000,003C,7580,3C00,0000,1222   ;1978  id[maint]_d  ; MAINT = 0
U 1222, 0000,003C,6580,3C00,0000,1223   ;1979  id[sbi.err]_d  ; sbi error reg = 0
U 1223, 0000,003C,7180,3C00,0000,1224   ;1980  id[comp]_d  ; comp reg = 0
U 1224, 0000,003C,E580,3C00,0000,1225   ;1981  ID[39]_d   ; Clear code for subroutine START.TEST
U 1225, 0001,003C,0180,09F0,0000,1226   ;1982  rc[0e]_d  ;
U 1226, 0001,003C,0180,0AF8,0000,1227   ;1983  r[0f]_d   ; Clear expected trap mask
U 1227, 0001,003C,0180,09F8,0000,105C   ;1984  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,124F   ;1985 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E   ;1986  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1987  .PAGE
            ;1988  .TOC "   Error Loop Routine"
            ;1989 ;++
            ;1990 ; FUNCTIONAL DESCRIPTION:
            ;1991 ;
            ;1992 ; This routine is called with the "ERLOOP" macro. The
            ;1993 ; routine calls the Micro Monitor to setup an error loop.
            ;1994 ;
            ;1995 ; CALLING CONSTRAINT:
            ;1996 ;
            ;1997 ; =0
            ;1998 ;
            ;1999 ; SIDE EFFECTS:
            ;2000 ;
            ;2001 ; D Reg - Destroyed
            ;2002 ;--
U 1228, 0818,0038,0980,0800,0000,105E  ;2003 ERLOOP: d_k[.2],j/callcon
```

```
            ;2004  .PAGE
            ;2005  .TOC "   Error 1 Routine"
            ;2006 ;++
            ;2007 ; FUNCTIONAL DESCRIPTION:
            ;2008 ;
            ;2009 ; This routine is used to report an error to the user. This
            ;2010 ; routine is used when you do not wish to continue in the
            ;2011 ; same test after the call to the Monitor.
            ;2012 ;
            ;2013 ; CALLING CONSTRAINT:
            ;2014 ;
            ;2015 ; None
            ;2016 ;
            ;2017 ; INPUTS:
            ;2018 ;
            ;2019 ; rc[0f]<15:8> - Contain the subtest number
            ;2020 ;
            ;2021 ; OUTPUTS:
            ;2022 ;
            ;2023 ; D<15:8> - Subtest number
            ;2024 ; D<7:0> - Error 1 Monitor Code
            ;2025 ;--
U 1229, 0810,0038,0180,0978,0000,122A  ;2026 ERROR1: d_rc[0f]  ; Get the subtest number
U 122A, 0819,0034,4D80,0800,0000,104E  ;2027  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2028 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2029  .PAGE
            ;2030  .TOC "   ERROR1 WITH PARAMETER"
            ;2031 ;++
            ;2032 ; FUNCTIONAL DESCRIPTION:
            ;2033 ;
            ;2034 ; This subroutine takes  as parameter the number of arguments
            ;2035 ; to be printed to the console in the case of an error logout.
            ;2036 ;
            ;2037 ; CALLING CONSTRAINT:
            ;2038 ;
            ;2039 ; =0
            ;2040 ;INPUTS:
            ;2041 ;
            ;2042 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2043 ;--
            ;2044 =0
            ;2045 ERR1.ARG:
U 109C. 0000,003D,0180,0800,0000,122D  ;2046  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 109D. 0000,003C,0180,0800,0000,1229  ;2047  J/ERROR1  ; Go to ERROR1
            ;2048 ;
            ;2049 ;
```

```
            ;2050  .PAGE
            ;2051  .TOC "   Error 2 Routine"
            ;2052 ;**
            ;2053 ; FUNCTIONAL DESCRIPTION:
            ;2054 ;
            ;2055 ; This routine is used  to report an error to the user. This
            ;2056 ; routine is used when you wish to continue in the same test
            ;2057 ;       after the call to the Monitor.
            ;2058 ;
            ;2059 ; CALLING CONSTRAINT:
            ;2060 ;
            ;2061 ; =0
            ;2062 ;
            ;2063 ; INPUTS:
            ;2064 ;
            ;2065 ; rc[0f]<15:8> - Contain the subtest number
            ;2066 ;
            ;2067 ; OUTPUTS:
            ;2068 ;
            ;2069 ; D<15:8> - Subtest number
            ;2070 ; D<7:0> - Error 2 Monitor Code
            ;2071 ;--
U 122B, 0810,0038,0180,0978,0000,122C  ;2072 ERROR2: d_rc[0f]  ; Get the subtest number
U 122C, 0819,0034,4D80,0800,0000,105A  ;2073  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2074 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2075 ;
```

```
            ;2076  .PAGE
            ;2077  .TOC "   ERROR 2 WITH PARAMETER"
            ;2078 ;++
            ;2079 ; FUNCTIONAL DESCRIPTION:
            ;2080 ;
            ;2081 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2082 ; except that in this case after setting the number of arguments
            ;2083 ; too the console, control is transferred to routine ERROR2.
            ;2084 ;
            ;2085 ; INPUTS:
            ;2086 ;
            ;2087 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2088 ;
            ;2089 ;--
            ;2090 =0
            ;2091 ERR2.ARG:
U 109E. 0000,003D,0180,0800,0000,122D  ;2092  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 109F. 0000,003C,0180,0800,0000,122B  ;2093  J/ERROR2  ; Go to ERROR2
            ;2094 ;
```

```
            ;2095  .PAGE
            ;2096  .TOC "   Micro-Monitor Call"
            ;2097 ;+
            ;2098 ; FUNCTIONAL DESCRIPTION:
            ;2099 ;
            ;2100 ; This micro word calls the micro monitor.
            ;2101 ;
            ;2102 ; EXPLICIT INPUTS:
            ;2103 ;
            ;2104 ; D reg must contain valid monitor code.
            ;2105 ;--
            ;2106 105E:
            ;2107 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2108  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2109  .PAGE
            ;2110  .TOC "   Reserved Control Store"
            ;2111 ;++
            ;2112 ; FUNCTIONAL DESCRIPTION:
            ;2113 ;
            ;2114 ; The following 16 locations must be reserved so the monitor
            ;2115 ; can use them to perform various functions that require
            ;2116 ; the execution of micro-code.
            ;2117 ;--
U 13F0,  0000,003C,0180,0B00,0000,13F1  ;2118 13F0: nop
U 13F1,  0000,003C,0180,0B00,0000,13F2  ;2119 13F1: nop
U 13F2,  0000,003C,0180,0B00,0000,13F3  ;2120 13F2: nop
U 13F3,  0000,003C,0180,0B00,0000,13F4  ;2121 13F3: nop
U 13F4,  0000,003C,0180,0B00,0000,13F5  ;2122 13F4: nop
U 13F5,  0000,003C,0180,0B00,0000,13F6  ;2123 13F5: nop
U 13F6,  0000,003C,0180,0B00,0000,13F7  ;2124 13F6: nop
U 13F7,  0000,003C,0180,0B00,0000,13F8  ;2125 13F7: nop
U 13F8,  0000,003C,0180,0B00,0000,13F9  ;2126 13F8: nop
U 13F9,  0000,003C,0180,0B00,0000,13FA  ;2127 13F9: nop
U 13FA,  0000,003C,0180,0B00,0000,13FB  ;2128 13FA: nop
U 13FB,  0000,003C,0180,0B00,0000,13FC  ;2129 13FB: nop
U 13FC,  0000,003C,0180,0B00,0000,13FD  ;2130 13FC: nop
U 13FD,  0000,003C,0180,0B00,0000,13FE  ;2131 13FD: nop
U 13FE,  0000,003C,0180,0B00,0000,13FF  ;2132 13FE: nop
U 13FF,  0000,003C,0180,0B00,0000,12AA  ;2133 13FF: nop
U 12AA,  0000,003C,0180,0B00,0000,122D  ;2134 12AA: nop ; This location is permanently blasted in the FPLA
            ;2135   ; to cause a micro ECO to location 12AA.
```

```
            ;2136  .PAGE
            ;2137  .TOC "   Set Argument Count"
            ;2138 ;++
            ;2139 ; FUNCTIONAL DESCRIPTION:
            ;2140 ;
            ;2141 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2142 ; to the console.
            ;2143 ;
            ;2144 ; CALLING CONSTRAINT:
            ;2145 ;
            ;2146 ; =0
            ;2147 ;
            ;2148 ; INPUTS:
            ;2149 ;
            ;2150 ; SC - Contains new value of argument count
            ;2151 ;
            ;2152 ; SIDE EFFECTS:
            ;2153 ;
            ;2154 ; Q is destroyed
            ;2155 ;--
U 122D, 0010,0038,01C0,0978,0000,122E  ;2156 SETRCF: q_rc[0f]  ; Get the argument count
U 122E, 0019,2034,4DC0,0800,0000,122F  ;2157  q_q.and.k[.ff00] ; ...
U 122F, 0019,2032,1D80,09F8,0000,0001  ;2158  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2159  .PAGE
            ;2160  .TOC "   Increment Subtest Number"
            ;2161 ;++
            ;2162 ; FUNCTIONAL DESCRIPTION:
            ;2163 ;
            ;2164 ; This routine is used to increment the subtest number
            ;2165 ; in RC[0F]<15:8>.
            ;2166 ;
            ;2167 ; CALLING CONSTRAINT:
            ;2168 ;
            ;2169 ; =0
            ;2170 ;
            ;2171 ; SIDE EFFECTS:
            ;2172 ;
            ;2173 ; D register is destroyed
            ;2174 ;--
            ;2175 subtst:
U 1230, 0810,0038,4D80,0978,0000,1231  ;2176  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2177  rc[0f]_d+k[.ff00], ; Increment and return
U 1231, 0019,4016,4D80,09F8,0000,0001  ;2178       word,return1 ; (Subtest number is negative)
```

```
            ;2179  .PAGE
            ;2180  .TOC " Diagnostic Specific Subroutines"
            ;2181  .TOC "   Select Controller"
            ;2182 ;**
            ;2183 ; FUNCTIONAL DESCRIPTION:
            ;2184 ;
            ;2185 ; This routine is used to put the memory system into the
            ;2186 ; specified configuration with respect to controller select.
            ;2187 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2188 ; a RETURN2 is made.
            ;2189 ;
            ;2190 ; CALLING CONSTRAINT:
            ;2191 ;
            ;2192 ; =00
            ;2193 ;
            ;2194 ; INPUTS:
            ;2195 ;
            ;2196 ; d - Contains value to write to bits<2:0> of Register A
            ;2197 ; rc[0e] - Address of Register A
            ;2198 ;
            ;2199 ; SIDE EFFECTS:
            ;2200 ;
            ;2201 ; sc,d,va are destroyed
            ;2202 ;--
            ;2203 SELECT.CNTRLR:
U 1232, 0010,0038,0180,0970,0284,7233  ;2204  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1233, 0801,001C,0180,0800,0000,1234  ;2205  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1234, 0000,003C,0180,A800,0000,1235  ;2206  cache.p_d[long]  ; Write the configuration Register
U 1235, 0818,0038,5D80,0800,0000,1236  ;2207  d_k[.7]  ; Get Misconfiguration bits
U 1236, 0000,003C,61F8,0800,0084,7237  ;2208  sc_k[.F],q_0  ; ...
U 1237, 0D00,003C,0180,0800,0000,1238  ;2209  d_dal.sc  ; ... D = x38000
U 1238, 0000,003C,01E0,0800,0000,1239  ;2210  q_d  ; Save mask
U 1239, 0000,003C,0180,C800,0000,123A  ;2211  d[long]_cache.p  ; Read CNFG A Reg and
            ;2212  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 123A, 001D,2034,0180,0800,0010,123B  ;2213  clk.ubcc ; ...
U 123B, 0000,013C,0180,0800,0000,10A0  ;2214  z?  ; Branch if no
U 10A0, 0000,003E,0180,0800,0000,0001  ;2215 =0 RETURN1  ; Bad configuration
U 10A1, 0000,003E,0180,0800,0000,0002  ;2216  RETURN2  ; Configuration ok
            ;2217
```

```
;2218  .PAGE
;2219  .TOC "   START TEST"
;2220 ;*************************************************************
;2221 ;**
;2222 ;
;2223 ; Functional Description:
;2224 ; ---------------------
;2225 ;
;2226 ;         This subroutine will be called by all tests that  check  the
;2227 ;    controllers,  or  those  tests  that  have  to switch between the
;2228 ;    controllers when executing.  Its main functions are:  select
;2229 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2230 ;    test that called it, call out an error  if  neither  one  of  the
;2231 ;    controllers can be configured properly, if all the controllers on
;2232 ;    a MS780-E/H were configured and tested, it should select the next
;2233 ;    MS780-E/H on the SBI and repeat the above sequence.
;2234 ;    A test making use of this  subroutine  should  be  configured  as
;2235 ;    follows:
;2236 ;
;2237 ;
;2238 ;        Begin TEST.XX
;2239 ;          :
;2240 ;          :   Call NEWTST
;2241 ;          :   Call FIND.MS780E
;2242 ;          :   IF No MS780-E's on the SBI
;2243 ;          :     THEN
;2244 ;          :         Skip to End of TEST.XX
;2245 ;          :     ELSE
;2246 ;          :
;2247 ;          : RESTART.TEST.XX:
;2248 ;          :
;2249 ;          :   Call START TEST
;2250 ;          :   IF Return1 from START.TEST
;2251 ;          :     THEN
;2252 ;          :         Skip to End of TEST.XX
;2253 ;          :     ELSE.
;2254 ;          :
;2255 ;          -
;2256 ;
;2257 ;          :
;2258 ;          :   Body of TEST.XX
;2259 ;          :
;2260 ;          -
;2261 ;
;2262 ;          :   Jump RESTART.TEST.XX
;2263 ;          :
;2264 ;        End TEST.XX
;2265 ;
;2266 ;
;2267 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2268 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2269 ;    should be passed when the subroutine is called a second, third or
;2270 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2271 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2272 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2273 ;    Register 39 can be interpreted as follows:
;2274 ;
;2275 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2276 ;               called the first time.  When this code is encountered
;2277 ;               this subroutine will try to select the lower controller.
;2278 ;               If the lower controller is misconfigured, the pointer in
;2279 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2280 ;               is re-entered.
;2281 ;               If, however there is no misconfiguration, the value in
;2282 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2283 ;               calling test.
;2284 ;
;2285 ;     b01   -   This value signifies that an attempt at configuring the
;2286 ;               lower controller failed, and the subroutine will attempt
;2287 ;               to select the upper controller.
;2288 ;               If the upper controller is also misconfigured execution
;2289 ;               stops with a call to ERROR1.
;2290 ;               If there is no misconfiguration on this controller, then
;2291 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2292 ;               made to the calling test.
;2293 ;
;2294 ;     b10   -   This value signifies that the lower controller was selected
;2295 ;               previously and the test was executed once.  If the
;2296 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2297 ;               39 is loaded with b11 and an attempt is made to select the
;2298 ;               upper controller.
;2299 ;               If there is a misconfiguration on this controller, this
;2300 ;               subroutine is just re-entered.
;2301 ;               Otherwise, a Return2 will be made to the calling test.
;2302 ;
;2303 ;     b11   -   This code identifies the cases in which the test has
;2304 ;               been executed at least once (i.e., at least one of the
;2305 ;               controllers on the MS780-E unit under test could be
;2306 ;               configured properly).  When this code is detected the
;2307 ;               subroutine clears ID Reg 39 and makes a call to
;2308 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2309 ;               tested then the subroutine executes a Return1.
;2310 ;               Otherwise the subroutine is re-entered.
;2311 ;
;2312 ;
;2313 ; Calling Constraint: =00
;2314 ; ------------------
;2315 ;
;2316 ;
;2317 ; Inputs:
;2318 ; ------
;2319 ;
;2320 ; ID Register 39 must be cleared by NEWTST
;2321 ;
;2322 ;
;2323 ; Outputs:
;2324 ; -------
;2325 ;
;2326 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2327 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2328 ;         to be tested
            ;2329 ;
            ;2330 ; Side Effects:
            ;2331 ; ------------
            ;2332 ;
            ;2333 ; Contents of the following regiters will be destroyed:
            ;2334 ;
            ;2335 ; D, Q
            ;2336 ;
            ;2337 ;--
            ;2338 ;*************************************************************
            ;2339 =0
            ;2340 START.TEST:
U 10A2, 0000,003D,0180,0800,0000,124A  ;2341  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 10A3, 0000,003C,0180,0800,0000,10A4  ;2342  NOP     ; ...tests that have more than one
            ;2343 =0      ; ...subtest.)
U 10A4, 0000,003D,0180,0800,0000,1228  ;2344  ERLOOP    ; Set up the error loop for the
            ;2345       ; ...eventuality that the MS780-E/H
            ;2346       ; ...currently under test has no
            ;2347       ; ...Controllers
            ;2348 RE.ENTRY:    ; Re entry point for this routine
U 10A5, 0000,003C,E5F0,2C00,0000,123C  ;2349  Q_ID[39]   ; Get the code
U 123C, 0C00,003C,0180,0800,0000,123D  ;2350  D_Q     ; Put it in the D Register
U 123D, 0000,193C,0180,0800,0000,100C  ;2351  D1-0?     ; Branch on the code
U 100C, 0000,003C,0180,0800,0000,1008  ;2352 =1100 J/STRT.CASE00   ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1010  ;2353  J/STRT.CASE01   ; Code is 01
U 100E, 0000,003C,0180,0800,0000,1242  ;2354  J/STRT.CASE10   ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1017  ;2355  J/STRT.CASE11   ; Code is 11
            ;2356 ;
            ;2357 ;
            ;2358 ; At this point the code in ID[39] was 00
            ;2359 ;
            ;2360 ;
            ;2361 =00
            ;2362 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1024  ;2363  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2364  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,123E  ;2365   D_K[.1]    ; Set up the code for re entry to this
            ;2366       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2367  D_K[.2]     ; Set up the code for a next call to
            ;2368       ; ...START.TEST indicating that the
            ;2369       ; ...Lower Controller was configured
            ;2370       ; ...( 10 )
            ;2371  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2372   RETURN2    ; Let the test know that the Lower
            ;2373       ; ...has been configured
            ;2374 STRT.LOW.MIS:
            ;2375  ID[39]_D,    ; Save code in ID[39] signifing that
U 123E, 0000,003C,E580,3C00,0000,10A5  ;2376   J/RE.ENTRY   ; ...the Lower Controller was Misconfig
            ;2377       ; ...and re enter this subroutine
            ;2378
            ;2379 ;
            ;2380 ;
            ;2381 ; At this point the code is 01
            ;2382 ;
```

```
            ;2383 ;
            ;2384 =00
            ;2385 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1028  ;2386  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2387  J/STRT.UPR.MIS,    ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,123F  ;2388   D_0    ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2389  D_K[.3]    ; Upper controller was configured
            ;2390       ; ...thus the code must be changed
            ;2391       ; ...accordingly ( 11 )
            ;2392  ID[39]_D,    ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2393   RETURN2    ; Let the test know that the Upper
            ;2394       ; ...Controller has been configured
            ;2395 STRT.UPR.MIS:
U 123F, 0000,003C,E580,3C00,0000,1240  ;2396  ID[39]_D    ; Save the Code ( 00 )
U 1240, 0018,0038,0D80,09E8,0000,1241  ;2397  RC[0D]_K[.3]    ; Set up a code for the Fail Chain
U 1241, 0000,003D,0980,0800,0084,709C  ;2398  ERROR1[.2]    ; Flag the error.
            ;2399 ;
            ;2400 ;
            ;2401 ; At this point the code is 10
            ;2402 ;
            ;2403 ;
            ;2404 STRT.CASE10:
U 1242, 0818,0038,0D80,0800,0000,1014  ;2405  D_K[.3]    ; Set the code to 11
            ;2406
            ;2407
            ;2408 =00 ID[39]_D,    ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1028  ;2409   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,10A5  ;2410  J/RE.ENTRY    ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2411  RETURN2    ; Upper Controller configured
            ;2412       ; ...let the test know it
            ;2413 ;
            ;2414 ;
            ;2415 ; At this point the code is 11
            ;2416 ;
            ;2417 ;
            ;2418 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1020  ;2419  D_0    ; Clear the code
            ;2420 =00 ID[39]_D,    ; Save the code
U 1020, 0000,003D,E580,3C00,0000,12A0  ;2421   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2422       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,10A5  ;2423  J/RE.ENTRY    ; Found another MS780-E/H.   Go and
            ;2424       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2425  RETURN1    ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1024  ;2426  NOP
```

```
;2427  .PAGE
;2428  .TOC "   SELECT LOWER CONTROLLER"
;2429 ;*************************************************************
;2430 ;++
;2431 ;
;2432 ; Functional Description:
;2433 ; ----------------------
;2434 ;
;2435 ; This subroutine can be called to select the lower
;2436 ; Controller on a MS780-E/H.
;2437 ; If the Lower controller is misconfigured then a
;2438 ; RETURN1 will be made.  Otherwise a RETURN2
;2439 ;
;2440 ; Calling Constraint: =00
;2441 ; -----------------
;2442 ;
;2443 ;
;2444 ; Inputs:
;2445 ; ------
;2446 ;
;2447 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2448 ;
;2449 ; Outputs:
;2450 ; -------
;2451 ;
;2452 ; RC[0D] will have the address of CNFG Register C
;2453 ;        ( 2000x008 )
;2454 ;
;2455 ; Side Effects:
;2456 ; ------------
;2457 ;
;2458 ; Contents of the following registers will lost:
;2459 ;
;2460 ; D
;2461 ;
;2462 ; The Lower controller on the MS780-E/H will be configured
;2463 ; when the subroutine is exitted with a RETURN2
;2464 ;--
;2465 ;*************************************************************
;2466 =00
;2467 SELECT.LOWER:
;2468  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,1232  ;2469   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2470  RETURN1    ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2471  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2472  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2473   RETURN2   ; Let caller know that the controller
;2474       ; ...was configured properly
```

```
;2475  .PAGE
;2476  .TOC "   SELECT UPPER CONTROLLER"
;2477 ;*************************************************************
;2478 ;++
;2479 ;
;2480 ; Functional Description:
;2481 ; ----------------------
;2482 ;
;2483 ; This subroutine can be called to select the upper
;2484 ; Controller on a MS780-E/H.
;2485 ; If the Upper controller is misconfigured then a
;2486 ; RETURN1 will be made.  Otherwise a RETURN2
;2487 ;
;2488 ; Calling Constraint: =00
;2489 ; ------------------
;2490 ;
;2491 ;
;2492 ; Inputs:
;2493 ; ------
;2494 ;
;2495 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2496 ;
;2497 ; Outputs:
;2498 ; -------
;2499 ;
;2500 ; RC[0D] will have the address of CNFG Register D
;2501 ;        ( 2000x010 )
;2502 ;
;2503 ; Side Effects:
;2504 ; ------------
;2505 ;
;2506 ; Contents of the following registers will lost:
;2507 ;
;2508 ; D
;2509 ;
;2510 ; The Upper controller on the MS780-E/H will be configured
;2511 ; when the subroutine is exitted with a RETURN2
;2512 ;--
;2513 ;*************************************************************
;2514 =00
;2515 SELECT.UPPER:
;2516  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,1232  ;2517   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2518  RETURN1     ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2519  D_RC[0E]    ; Get MS780-E/H I/O Base address
        ;2520  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2521   RETURN2    ; Let caller know that the controller
        ;2522       ; ...was configured properly
```

```
            ;2523  .PAGE
            ;2524  .TOC "    Set Diagnostic Mode Routine"
            ;2525 ;++
            ;2526 ; FUNCTIONAL DESCRIPTION:
            ;2527 ;
            ;2528 ; This routine is used to set the specified diagnostic mode
            ;2529 ; in Register B. Other bits in the register remain unchanged.
            ;2530 ;
            ;2531 ; CALLING CONSTRAINT:
            ;2532 ;
            ;2533 ; =0
            ;2534 ;
            ;2535 ; INPUTS:
            ;2536 ;
            ;2537 ; d - Contains the mode to set in bits<1:0>
            ;2538 ; rc[0e] - Contains base address of controller
            ;2539 ;
            ;2540 ; SIDE EFFECTS:
            ;2541 ;
            ;2542 ; d,va,sc are destroyed
            ;2543 ;--
            ;2544 SET_DIAG_MODE:
U 1243, 0010,0038,D9F8,0970,0284,7244  ;2545  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 1244, 0D00,003C,0180,0803,0000,1245  ;2546  va_va+4,d_dal.sc ; Shift specified mode into position
U 1245, 0000,003C,01E0,0800,0000,1246  ;2547  q_d   ; Save the diagnostic mode bits
U 1246, 0000,003C,0180,C800,0000,1247  ;2548  d[long]_cache.p  ; Read the contents of the CNFG register B
U 1247, 081D,0030,0180,0800,0000,1248  ;2549  d_d.or.q  ; Set the diagnostic mode bits
U 1248, 0000,003E,0180,A800,0000,0001  ;2550  cache.p_d[long],return1 ; Set the specified mode and return
            ;2551
```

```
            ;2552  .PAGE
            ;2553  .TOC "   SBI Unjam Routine"
            ;2554 ;**
            ;2555 ; FUNCTIONAL DESCRIPTION:
            ;2556 ;
            ;2557 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2558 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2559 ; and finally HOLD for the last 16 cycles.
            ;2560 ;
            ;2561 ; CALLING CONSTRAINT:
            ;2562 ;
            ;2563 ; =0
            ;2564 ;
            ;2565 ; SIDE EFFECTS:
            ;2566 ;
            ;2567 ; D Reg destroyed
            ;2568 ;--
            ;2569    ;
            ;2570    ; assert hold for 16 cycles
            ;2571    ;
            ;2572 SBI.UNJAM:
            ;2573  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 1249, 0818,0038,6580,8000,0010,10A6  ;2574       clk.ubcc  ; set micro cc's for next cycle
            ;2575 =0
            ;2576 SBI.UNJAM.1:
            ;2577  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2578       clk.ubcc,z?, ; ...
U 10A6, 0819,0100,0580,8000,0010,10A6  ;2579       j/sbi.unjam.1 ; test for completion
            ;2580    ;
            ;2581    ; assert hold and unjam for 16 cycles
            ;2582    ;
            ;2583  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 10A7, 0818,0038,6580,8800,0010,10A8  ;2584       d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2585 =0
            ;2586 SBI.UNJAM.2:
            ;2587  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2588       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 10A8, 0819,0100,0580,8800,0010,10A8  ;2589       z?,j/sbi.unjam.2 ; ...
            ;2590    ;
            ;2591    ; assert hold for 16 cycles
            ;2592    ;
            ;2593  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 10A9, 0818,0038,6580,8000,0010,10AA  ;2594       clk.ubcc  ; and set cc's for next cycle
            ;2595 =0
            ;2596 SBI.UNJAM.3:
            ;2597  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2598       clk.ubcc,z?, ; ...
U 10AA, 0819,0100,0580,8000,0010,10AA  ;2599       j/sbi.unjam.3 ; ...
U 10AB, 0000,003E,0180,0800,0000,0001  ;2600  return1   ; exit
```

```
            ;2601  .PAGE
            ;2602  .TOC "   RESET SUBTEST NUMBER"
            ;2603 ;++
            ;2604 ; FUNCTIONAL DESCRIPTION:
            ;2605 ;
            ;2606 ; Since some  of the tests will have to be restarted when two
            ;2607 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2608 ; the subtest counter to zero ( only for thoes tests that have
            ;2609 ; more than one subtest). This subroutine may also be necessary
            ;2610 ; when a test is being loop on.
            ;2611 ;
            ;2612 ; CALLING CONSTRAINT:
            ;2613 ;
            ;2614 ; =0
            ;2615 ; SIDE EFFECTS:
            ;2616 ;
            ;2617 ; D is destroyed
            ;2618 ;
            ;2619 ;--
            ;2620 RESET.SUBTST:
            ;2621  RC[0F]_0,   ; Reset subtest number
U 124A, 0003,003E,0180,09F8,0000,0001  ;2622   RETURN1   ; ...and return
```

```
            ;2623  .PAGE
            ;2624  .TOC "   Initialize the SBI"
            ;2625 ;++
            ;2626 ; FUNCTIONAL DESCRIPTION:
            ;2627 ;
            ;2628 ; This routine performs the following functions:
            ;2629 ;
            ;2630 ;  Executes an SBI Unjam
            ;2631 ;  Clears the SBI Fault Latch
            ;2632 ;  Clears the SBI Error Register
            ;2633 ;
            ;2634 ; CALLING CONSTRAINT:
            ;2635 ;
            ;2636 ; =0
            ;2637 ;
            ;2638 ; SIDE EFFECTS:
            ;2639 ;
            ;2640 ; D & Q Register destroyed
            ;2641 ;--
            ;2642 =0
            ;2643 SBI.INIT:
U 10AC, 0000,003D,0180,0800,0000,1249  ;2644  call[sbi.unjam]  ; Unjam the SBI
U 10AD, 0818,0038,C180,0800,0000,124B  ;2645  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 124B, 0801,0028,6580,3C00,0000,124C  ;2646  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 124C, 0F00,003C,6D80,3C00,0000,124D  ;2647  id[fault]_d,d_0
U 124D, 0000,003C,6D80,3C00,0000,124E  ;2648  id[fault]_d
U 124E, 0000,003E,6580,3C00,0000,0001  ;2649  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2650  .PAGE
            ;2651  .TOC "   Disable Cache Routine"
            ;2652 ;++
            ;2653 ; FUNCTIONAL DESCRIPTION:
            ;2654 ;
            ;2655 ; This routine disables cache hits by setting bits <16:15>
            ;2656 ; in the maintenance register.
            ;2657 ;
            ;2658 ; CALLING SEQUENCE:
            ;2659 ;
            ;2660 ; =0
            ;2661 ;
            ;2662 ; SIDE EFFECTS:
            ;2663 ;
            ;2664 ; D & Q Registers destroyed
            ;2665 ;--
            ;2666 DISABLE.CACHE:
U 124F, 0818.0038.4580.0800.0000.1250  ;2667  d_k[.8000]  ; ... and seed constant
U 1250, 0500.003C.0180.0800.0000.1251  ;2668  d_d.left  ; Generate final constant
U 1251, 0819.0030.4580.0800.0000.1252  ;2669  d_d.or.k[.8000]  ; ... = 00018000
U 1252, 0000.003E.7580.3C00.0000.0001  ;2670  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2671  .PAGE
            ;2672  .TOC "   Enable Cache"
            ;2673 ;++
            ;2674 ; FUNCTIONAL DESCRIPTION:
            ;2675 ;
            ;2676 ; This routine enables cache group 0 by clearing the force
            ;2677 ; miss bit in the maintenance register.
            ;2678 ;
            ;2679 ; CALLING CONSTRAINT:
            ;2680 ;
            ;2681 ; =0
            ;2682 ;
            ;2683 ; SIDE EFFECTS:
            ;2684 ;
            ;2685 ; D, Q AND SC Registers destroyed
            ;2686 ;--
            ;2687 ENABLE.CACHE:
U 1253, 0000,003C,75F0,2C00,0000,1254  ;2688  q_id[maint]  ; Get current maintenance register
U 1254, 0000,003C,6580,0800,0084,7255  ;2689  sc_k[.10]  ; Setup to clear bit 16
U 1255, 0801,2034,0180,0800,0000,1256  ;2690  d_q.and.mask  ; Clear bit 16
U 1256, 0000,003E,7580,3C00,0000,0001  ;2691  id[maint]_d,return1 ; Rewrite register and return
```

```
          ;2692  .PAGE
          ;2693  .TOC "   Wait Loop Routine"
          ;2694 ;**
          ;2695 ; FUNCTIONAL DESCRIPTION:
          ;2696 ;
          ;2697 ; As the name implies,  this subroutine is used to set up a wait
          ;2698 ; loop for a specified  number of cycles.  This may be necessary
          ;2699 ; when the micro-program has to wait for another event to finish.
          ;2700 ; When the subroutine is entered Register D should be holding the
          ;2701 ; desired numbered of cycles.
          ;2702 ;
          ;2703 ; CALLING CONSTRAINT:
          ;2704 ;
          ;2705 ; =0
          ;2706 ;
          ;2707 ; INPUTS:
          ;2708 ;
          ;2709 ; d - Contains number of cycles to wait
          ;2710 ; alu.z condition code must not be set
          ;2711 ;
          ;2712 ; SIDE EFFECTS:
          ;2713 ;
          ;2714 ; d is destroyed
          ;2715 ;--
          ;2716 WAIT.LOOP:
U 1257, 0018,0038,6180,0800,0010,10AE  ;2717  alu_k[.F],clk.ubcc ; Make sure alu.z condition
          ;2718      ; ...is not set
          ;2719 =0
          ;2720 W.LOOP:
          ;2721  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10AE, 0819,0100,0580,0800,0010,10AE  ;2722       j/W.LOOP
U 10AF, 0000,003E,0180,0800,0000,0001  ;2723  return1    ; exit
          ;2724
```

```
            ;2725  .PAGE
            ;2726  .TOC "   Check for Interrupt Routine"
            ;2727 ;++
            ;2728 ; FUNCTIONAL DESCRIPTION:
            ;2729 ;
            ;2730 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2731 ; If an interrupt is active, the following registers are setup:
            ;2732 ;  RC[8] - Gets the VECTOR register
            ;2733 ;  RC[7] - Gets the SBI ERROR register
            ;2734 ;  RC[6] - Gets the FAULT register
            ;2735 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2736 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2737 ;
            ;2738 ; CALLING CONSTRAINT:
            ;2739 ;
            ;2740 ; =00
            ;2741 ;
            ;2742 ; SIDE EFFECTS:
            ;2743 ;
            ;2744 ; D & Q are destroyed
            ;2745 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2746 ;--
            ;2747 CHECK_INTERRUPT:
            ;2748    ;
            ;2749    ; First, enable the fault and RDS/CRD interrupts
            ;2750    ;
U 1258, 0818,0038,4580,0800,0000,1259  ;2751  d_k[.8000]  ; Get data to set interrupt enable
            ;2752  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1259, 0500,003C,6580,3C00,0000,125A  ;2753       d_d.left
U 125A, 0100,003C,0180,0800,0000,125B  ;2754  d_d.left2  ; D = 40000
U 125B, 0000,003C,6D80,3C00,0000,125C  ;2755  id[fault]_d  ; Set fault interrupt enable
            ;2756  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 125C, 0F03,003C,0180,09A8,4000,125D  ;2757       rc[5]_0  ; and clear interrupt occurred flag
U 125D, 0000,003C,3D80,3C00,0000,125E  ;2758  id[psl]_d  ; Clear the PSL
U 125E, 0000,003C,6580,3C00,0000,125F  ;2759  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 125F, 0000,003C,6D80,3C00,0000,1260  ;2760  id[fault]_d  ; Clear FAULT Interrupt Enable
U 1260, 0000,0E3C,0180,0800,0000,1004  ;2761  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2762 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2763  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2764  return1   ; No interrupt
            ;2765 =111
            ;2766 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,1261  ;2767  q_id[vector]  ; Get ID Vector Register
U 1261, 0001,203C,65F0,2DC0,0000,1262  ;2768  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 1262, 0001,203C,6DF0,2DB8,0000,1263  ;2769  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 1263, 0001,203C,0180,09B0,0000,1264  ;2770  rc[6]_q   ; Save falut reg
U 1264, 0018,003A,0580,09A8,0000,0002  ;2771  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2772  .PAGE
            ;2773  .TOC "   CHECK UTRAP"
            ;2774 ;++
            ;2775 ;FUNCTIONAL DESCRIPTION:
            ;2776 ;
            ;2777 ; This subroutine is used to check wether a Utrap occured.
            ;2778 ; It takes the contents of the Save Utrap flags register
            ;2779 ; R[0E], and compares them to the contrnts of the Utrap
            ;2780 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2781 ; according to the results of the test  (i.e., contents of
            ;2782 ; these registers are equal or not).
            ;2783 ; CALLING CONSTRAINT:
            ;2784 ;
            ;2785 ; =00
            ;2786 ;
            ;2787 ; INPUTS:
            ;2788 ;
            ;2789 ; R[0E]- Saved Utrap Mask
            ;2790 ; R[0F]- Expected Utrap Mask
            ;2791 ;
            ;2792 ;--
            ;2793 CHECK_UTRAP:
U 1265. 0800,003C,0180,0A70,0000,1266  ;2794  D_R[0E]    ; Reg D is loaded with the trap mask
U 1266. 0001,003C,0180,09C0,0000,1267  ;2795  RC[08]_D   ; Save expected Utrap flag for error logout
U 1267. 0800,003C,0180,0A78,0000,1268  ;2796  D_R[0F]    ; Get recieved Utrap flag to D reg
U 1268. 0001,003C,0180,09B8,0000,1269  ;2797  RC[07]_D   ; Save in RC[07]
            ;2798  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 1269. 000D,0020,0180,0A70,0010,126A  ;2799          CLK.UBCC
U 126A. 0003,013C,0180,0AF8,0000,10B0  ;2800  Z?,R[0F]_0   ; Branch if no traps
U 10B0. 0000,003E,0180,0800,0000,0002  ;2801 =0 RETURN2    ; Utrap occured
U 10B1. 0000,003E,0180,0800,0000,0001  ;2802  RETURN1    ; No Utrap  return
            ;2803 ;
```

```
            ;2804  .PAGE
            ;2805  .TOC "   Set Trap Mask"
            ;2806 ;**
            ;2807 ; FUNCTIONAL DESCRIPTION:
            ;2808 ;
            ;2809 ; This routine is used to set a bit in the Micro Trap Mask
            ;2810 ; R[0F].
            ;2811 ;
            ;2812 ; CALLING CONSTRAINT:
            ;2813 ;
            ;2814 ; =0
            ;2815 ;
            ;2816 ; INPUTS:
            ;2817 ;
            ;2818 ; SC - Contains bit number to set.
            ;2819 ;
            ;2820 ; OUTPUTS:
            ;2821 ;
            ;2822 ; rc[0E] - Contains the current trap mask
            ;2823 ;
            ;2824 ; SIDE EFFECTS:
            ;2825 ;
            ;2826 ; d regsiter is destroyed
            ;2827 ;--
            ;2828 SET_TRAP_MASK:
U 126B, 0800,003C,0180,0A78,0000,126C  ;2829  d_r[0f]
U 126C, 0801,001C,0180,0AF8,0000,126D  ;2830  r[0f]_d.ornot.mask,d_alu ; Set mask
U 126D, 0001,003E,0180,0AF0,0000,0001  ;2831  r[0E]_d,return1    ; Save mask and return
```

```
            ;2832  .PAGE
            ;2833  .TOC "   FIND MS780-E MEMORY"
            ;2834 ;++
            ;2835 ; FUNCTIONAL DESCRIPTION:
            ;2836 ;
            ;2837 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2838 ;      MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2839 ;      for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2840 ;      3D will hold the I/O base address of the MS780-E subsystems found
            ;2841 ;      on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2842 ;      MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2843 ;      first MS780-E found will have  its  starting  address  set  to  0
            ;2844 ;      (zero).
            ;2845 ;      All the other MS780-E/H's found will have their starting  address
            ;2846 ;      set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2847 ;      Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2848 ;      apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2849 ;      current MS780-E unit under test.
            ;2850 ;      If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2851 ;      set to maximum.
            ;2852 ;
            ;2853 ; CALLING CONSTRAINT:
            ;2854 ;
            ;2855 ; =00
            ;2856 ;
            ;2857 ; OUTPUTS:
            ;2858 ;
            ;2859 ; rc[0e] - Contains address of Configuration Register
            ;2860 ; r[0]  - Contains TR level
            ;2861 ;
            ;2862 ; SIDE EFFECTS:
            ;2863 ;
            ;2864 ; D register is destroyed.
            ;2865 ;--
            ;2866 =0
            ;2867 FIND.MS780E:
U 10B2. 000D,003D,0180,0800,0000,10AC  ;2868  call[sbi.init]  ; Make sure SBI is enabled
U 10B3. 0003,003C,0180,0988,0000,126E  ;2869  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 126E. 081B,0038,1980,0800,0000,126F  ;2870  d_k[zero]  ; Clear MS780-E/H counter
U 126F. 000D,003C,F980,3C00,0000,1270  ;2871  id[3e]_d  ; ...
U 1270. 001B,0038,0580,0A80,0000,1271  ;2872  r[0]_k[.1]  ; Init TR counter
U 1271. 0F03,003C,0180,09F0,0000,10B4  ;2873  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2874      ; ...CNFG A Reg
            ;2875 =0
            ;2876 FIND.MEM.LOOP:
U 10B4. 080D,003D,0180,0A00,0000,1297  ;2877  d_r[0],call[generate.io]; Generate address of TR level
U 10B5. 0001,003C,0180,09F0,0200,10B6  ;2878  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10B6. 000D,003D,8980,0800,0084,726B  ;2879 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2880  d[long]_cache.p, ; Read the specified tr level
U 10B7. 0000,003C,0180,C970,0000,1272  ;2881    ic_rc[0e]  ; ...
U 1272. 0000,003C,0180,0A78,0010,1273  ;2882  alu_r[0f],clk.ubcc ; Check for no response
U 1273. 0001,013C,0180,0880,0082,10B8  ;2883  z?,sc_d,la_ra[0] ; Branch if no adapter here
U 10B8. 0000,003C,0180,0800,0000,1274  ;2884 =0 j/check.adpt.code ; Check for a memory
U 10B9. 0000,003C,0180,0800,0000,1293  ;2885  j/find.mem.lp1  ; Try next tr
            ;2886 CHECK.ADPT.CODE:
```

```
U 1274, 0000,003C,1D80,0800,1404,7275  ;2887  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 1275, 0000,163C,0180,0800,0000,1018  ;2888  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,7276  ;2889 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,7277  ;2890  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,1293  ;2891  j/find.mem.lp1  ; Not a memory
U 101B, 0000,003C,0180,0800,0000,1293  ;2892  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,727C  ;2893  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,1293  ;2894  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,1280  ;2895  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,1280  ;2896  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2897   ;
          ;2898   ; Found an MS780-A or -C. Set the starting address
          ;2899   ; to maximum.
          ;2900   ;
U 1276, 0818,0038,5D80,0800,0000,1278  ;2901 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1277, 0818,0038,0580,0800,0000,1278  ;2902 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
          ;2903 MS780.COMMON:
U 1278, 0819,0030,7180,0800,0000,1279  ;2904  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1279, 0000,003C,01F8,0803,0080,D27A  ;2905  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 127A, 0D00,003C,0180,0800,0000,127B  ;2906  d_dal.sc  ; Put data in bits<29:14>
          ;2907  cache.p_d[long], ; Set the starting address to maximum
U 127B, 0000,003C,0180,A800,0000,1293  ;2908       j/find.mem.lp1 ; and continue TR scan
          ;2909   ;
          ;2910   ; Found an MA780. Set the starting address to maximum
          ;2911   ;
          ;2912 MA780.MAX:
U 127C, 0818,0038,39F8,0803,0000,127D  ;2913  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 127D, 0D00,003C,0180,0803,0000,127E  ;2914  d_dal.sc,va_va+4 ; Put in proper position
U 127E, 0000,003C,0180,0803,0000,127F  ;2915  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;2916  cache.p_d[long], ; Set starting address to maximum
U 127F, 0000,003C,0180,A800,0000,1293  ;2917       j/find.mem.lp1
          ;2918 ;
          ;2919 ; Found an MS780E
          ;2920 ;
          ;2921 FOUND.MS780E:
U 1280, 0000,003C,F9F0,2C00,0000,1281  ;2922  q_id[3e]  ; Set up to see how many MS780-E's
          ;2923  alu_q.xor.k[.3], ; Are there Maximum units?
U 1281, 0019,2020,0D80,0800,0010,1282  ;2924   clk.ubcc ; Check condition codes
U 1282, 0000,013C,0180,0800,0000,10BA  ;2925  z?   ; Are we at maximum units for system
U 10BA, 0000,003C,0180,0800,0000,1283  ;2926 =0 J/ms780e.tst  ; No go find how many units ther are
U 10BB, 0818,0038,C180,0800,0000,10BC  ;2927  d_k[.ffff]  ; Yes set address to maximum
U 10BC, 0000,003D,0180,0800,0000,129B  ;2928 =0 call[set.start.addr] ; .....
U 10BD, 0000,003C,0180,0800,0000,1293  ;2929  j/find.mem.lp1  ; Go check to see if we are done
          ;2930 ;
          ;2931 MS780E.TST:
          ;2932  alu_rc[01],  ; Check "Found MS780E Flag"
U 1283, 0010,0038,0180,0908,0010,1284  ;2933   clk.ubcc ; Check condition codes
U 1284, 0810,0138,0180,0970,0000,10BE  ;2934  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;2935 =0 j/not.first.ms780e, ; No
U 10BE, 0818,0038,C180,0800,0000,10C2  ;2936   d_k[.ffff] ; Set starting address
U 10BF, 0001,003C,0180,0988,0000,1285  ;2937  rc[01]_d  ; Yes put base address in rc[01]
U 1285, 0818,0038,7980,0800,0000,1286  ;2938  d_k[.30]  ; Set up SC to point to first unit
U 1286, 0019,0014,F580,0800,0082,1287  ;2939  alu_d+k[.a],sc_alu ; ...
U 1287, 0810,0038,0180,0908,0000,1288  ;2940  d_rc[01]  ; Put base address of unit in D
U 1288, 0000,003C,0180,3400,0000,1289  ;2941  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1289, 0F00.003C.0180.0800.0000.10C0  ;2942  d_0   ; Prepare to set starting address to Zero
U 10C0, 0000.003D.0180.0800.0000.129B  ;2943 =0 call[set.start.addr] ; Go do it
        ;2944  j/find.mem.lp1,  ; Check to see if we are done
U 10C1, 0003.003C.0180.09E8.0000.1293  ;2945   rc[0d]_0 ; ....
        ;2946 =0
        ;2947 NOT.FIRST.MS780E:
U 10C2, 0000.003D.0180.0800.0000.129B  ;2948  call[set.start.addr] ; Go set starting address to maximum
U 10C3, 0000.003C.F9F0.2C00.0000.128A  ;2949  q_id[3e]  ; Get the number of MS780-Es found
U 128A, 0819.2014.0580.0800.0000.128B  ;2950  d_q+k[.1]  ; Increment this counter
U 128B, 0000.003C.F980.3C00.0000.128C  ;2951  id[3e]_d  ; Save the counter
U 128C, 0018.0038.11C0.0800.0000.128D  ;2952  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2953     ; ...were the I/O base addresses will be stored
U 128D, 081D.2000.7980.0800.0000.128E  ;2954  d_q-d,k[.30]  ; ... adjust pointer
U 128E, 0819.0014.7980.0800.0000.128F  ;2955  d_d+k[.30]  ; Point the SC to the ID register
U 128F, 0000.003C.F580.0800.0000.1290  ;2956  k[.a]   ; ...to receive I/O base address
U 1290, 0019.0014.F580.0800.0082.1291  ;2957  alu_d+k[.a],sc_alu ; ...
U 1291, 0810.0038.0180.0970.0000.1292  ;2958  d_rc[0e]  ; Get base address to d
U 1292, 0000.003C.0180.3400.0000.1293  ;2959  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2960   ;
        ;2961   ; Try next tr
        ;2962   ;
        ;2963 FIND.MEM.LP1:
U 1293, 0818.0014.0580.0A80.0000.1294  ;2964  r[0]_la+k[.1],d_alu ; Increment TR level
        ;2965  alu_d.and.k[.f],
U 1294, 0019.0034.6180.0800.0010.1295  ;2966      clk.ubcc  ; Check if all done
U 1295, 0000.013C.0180.0800.0000.10C4  ;2967  z?   ; Branch if yes
U 10C4, 0000.003C.0180.0800.0000.10B4  ;2968 =0 j/find.mem.loop  ; Try next TR
U 10C5, 0810.0038.0180.0908.0010.1296  ;2969  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1296, 0000.013C.0180.0800.0000.10C6  ;2970  z?   ; Check condition codes
U 10C6, 0001.003E.0180.09F0.0000.0002  ;2971 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10C7, 0000.003E.0180.0800.0000.0001  ;2972  return1   ; No MS780E found
        ;2973
```

```
            ;2974  .PAGE
            ;2975  .TOC "   Generate IO Address"
            ;2976 ;++
            ;2977 ; FUNCTIONAL DESCRIPTION:
            ;2978 ;
            ;2979 ; This routine is used to generate an SBI Configuration Register
            ;2980 ; address from a TR level.
            ;2981 ;
            ;2982 ; CALLING CONSTRAINT:
            ;2983 ;
            ;2984 ; =0
            ;2985 ;
            ;2986 ; INPUTS:
            ;2987 ;
            ;2988 ; D - Contains TR level
            ;2989 ;
            ;2990 ; OUTPUTS:
            ;2991 ;
            ;2992 ; D - Contains SBI IO address
            ;2993 ;--
            ;2994 GENERATE.IO:
U 1297, 0000.003C.89F8.0800.0084.7298  ;2995  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1298, 0D00.003C.8D80.0800.0084.7299  ;2996  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 1299, 0000.003C.0980.0800.0084.B29A  ;2997  sc_sc-k[.2]  ; insert bit 29
U 129A, 0801.001E.0180.0800.0000.0001  ;2998  d_d.ornot.mask,return1 ; and return
```

```
            ;2999  .PAGE
            ;3000  .TOC "   Set Starting Address"
            ;3001 ;++
            ;3002 ; FUNCTIONAL DESCRIPTION:
            ;3003 ;
            ;3004 ; This routine is used to set the starting address of the
            ;3005 ; memory to the specified value.
            ;3006 ;
            ;3007 ; CALLING CONSTRAINT:
            ;3008 ;
            ;3009 ; =0
            ;3010 ;
            ;3011 ; INPUTS:
            ;3012 ;
            ;3013 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3014 ;     (B Register Image divided by 2**16)
            ;3015 ; rc[0e] - Contains Address of Register A
            ;3016 ;
            ;3017 ; OUTPUTS:
            ;3018 ;
            ;3019 ; d - Contains aligned starting address (B Register Image)
            ;3020 ;
            ;3021 ; SIDE EFFECTS:
            ;3022 ;
            ;3023 ; d,q,va, and sc are destroyed
            ;3024 ;--
            ;3025 SET.START.ADDR:
U 129B, 0010,0038,6580,0970,0284,729C  ;3026  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 129C, 0000,003C,01F8,0803,0000,129D  ;3027  va_va+4,q_0  ; Set address to Register B
            ;3028  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 129D, 0D00,003C,0980,0800,0084,B29E  ;3029      sc_sc-k[.2]  ; Setup to insert bit 14
U 129E, 0801,001C,0180,0800,0000,129F  ;3030  d_d.ornot.mask  ; Insert Write Enable bit
U 129F, 0000,003E,0180,A800,0000,0001  ;3031  cache.p_d[long],return1 ; Set the starting address and return
            ;3032
```

```
            ;3033  .PAGE
            ;3034  .TOC "   ENABLE NEXT MS780-E"
            ;3035 ;++
            ;3036 ; FUNCTIONAL DESCRIPTION:
            ;3037 ;
            ;3038 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3039 ;    They will be tested  separately, according to the TR level at which
            ;3040 ;    they are located,  starting with the one  at the lowest level first.
            ;3041 ;          When  a test has finished  with the first unit, it will have to call
            ;3042 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3043 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3044 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3045 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3046 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3047 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3048 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3049 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3050 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3051 ;
            ;3052 ; CALLING CONSTRAINT:
            ;3053 ;
            ;3054 ;  =00
            ;3055 ;
            ;3056 ;--
            ;3057 ENABLE.NEXT.MS780E:
U 12A0, 0000,003C,F9F0,2C00,0000,12A1  ;3058  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3059  ALU_Q,    ; Check to see if it is zero
U 12A1, 0001,203C,0180,0800,0010,12A2  ;3060      CLK.UBCC  ; Check Condition Codes
U 12A2, 0000,013C,0180,0800,0000,10C8  ;3061  Z?    ; ...for zero
U 10C8, 0000,003C,0180,0800,0000,12A3  ;3062 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10C9, 0000,003E,0180,0800,0000,0002  ;3063  RETURN2    ; Zero No more units to test
            ;3064 ENABLE.NEXT:
U 12A3, 0818,0038,C180,0800,0000,10CA  ;3065  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10CA, 0000,003D,0180,0800,0000,129B  ;3066 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10CB, 0818,0038,7980,0800,0000,12A4  ;3067  D_K[.30]  ; Prepare to point to the ID register holding
            ;3068      ; ...the I/O Base address of the next MS780-E
U 12A4, 0819,0014,1180,0800,0000,12A5  ;3069  D_D+K[.4]  ; ...
U 12A5, 0000,003C,F580,0800,0000,12A6  ;3070  K[.A]    ; ...
U 12A6, 0819,0014,F580,0800,0000,12A7  ;3071  D_D+K[.A]  ; ...
U 12A7, 0000,003C,F9F0,2C00,0000,12A8  ;3072  Q_ID[3E]  ; Get MS780-E Counter
            ;3073  D_D-Q,    ; D now holds the pointer to the ID register
U 12A8, 081D,0000,0180,0800,0082,12A9  ;3074    SC_ALU  ; ...
U 12A9, 0000,003C,01F0,2400,0000,12AB  ;3075  Q_ID(SC)  ; Q gets adrress of next MS780E
U 12AB, 0001,203C,0180,09F0,0000,12AC  ;3076  RC[0E]_Q  ; Save it also in RC[0E]
U 12AC, 0F03,003C,0180,09E8,0000,10CC  ;3077  RC[0D]_0,D_0  ; Set starting address to zero
U 10CC, 0000,003D,0180,0800,0000,129B  ;3078 =0 CALL[SET.START.ADDR] ; ....
U 10CD, 0F00,003C,F9F0,2C00,0000,12AD  ;3079  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 12AD, 081D,2008,0180,0800,0000,12AE  ;3080  D_Q-D-1    ; Subtract 1 from total
U 12AE, 0000,003C,F980,3C00,0000,10CE  ;3081  ID[3E]_D  ; Adjust the pointer
U 10CE, 0000,003D,0180,0800,0000,124A  ;3082 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10CF, 0000,003E,0180,0800,0000,0001  ;3083  RETURN1    ; Tell caller another unit was found
```

```
            ;3084  .PAGE
            ;3085  .TOC "   Set ECC Bits"
            ;3086 ;++
            ;3087 ; FUNCTIONAL DESCRIPTION:
            ;3088 ;
            ;3089 ; This routine is used to set the specified ECC bits
            ;3090 ; in Register B. Other bits in the register remain unchanged.
            ;3091 ;
            ;3092 ; CALLING CONSTRAINT:
            ;3093 ;
            ;3094 ; =0
            ;3095 ;
            ;3096 ; INPUTS:
            ;3097 ;
            ;3098 ; d - Contains the bits to set in <6:0>
            ;3099 ; rc[0] - Contains base address of controller
            ;3100 ;
            ;3101 ; SIDE EFFECTS:
            ;3102 ;
            ;3103 ; d,va,sc are destroyed
            ;3104 ;--
            ;3105 SET_ECC:
U 12AF. 0010,0038,0180,0970,0200,12B0  ;3106  va_rc[0E]  ; Set address of Register B
U 12B0. 0000,003C,0180,0803,0000,12B1  ;3107  va_va+4    ; ...in VA
U 12B1. 0000,003C,01E0,C800,0000,12B2  ;3108  q_d,d[long]_cache.p ; Read current contents of register
U 12B2. 0819,0024,5980,0800,0000,12B3  ;3109  d_d.andnot.k[.7f] ; Clear current ECC bits
U 12B3. 081D,0030,0180,0800,0000,12B4  ;3110  d_d.or.q  ; Insert new ecc bits
U 12B4. 0000,003E,0180,A800,0000,0001  ;3111  cache.p_d[long],return1 ; Set the specified bits and return
            ;3112
            ;3113
```

```
                ;3114  .PAGE
                ;3115  .TOC "   MS780-E/H CNFG REG CLEANUP"
                ;3116 ;*****************************************************
                ;3117 ;<<
                ;3118 ;
                ;3119 ; Functional Description:
                ;3120 ; ----------------------
                ;3121 ;
                ;3122 ;      This subroutine is used to clear all error conditions
                ;3123 ; in the MS780-E/H Configuration/Status/Error Registers.
                ;3124 ; Bits beeing cleared are:
                ;3125 ;
                ;3126 ; CNFG Regsiter B <11:00>
                ;3127 ;
                ;3128 ; CNFG Register C and D <31:28> and <10:06>
                ;3129 ;   ---
                ;3130 ;
                ;3131 ;
                ;3132 ; Calling Constraint: =0
                ;3133 ; ------------------
                ;3134 ;
                ;3135 ;
                ;3136 ; Inputs:
                ;3137 ; ------
                ;3138 ;
                ;3139 ; RC[0E] MUST contain the I/O base address of
                ;3140 ; the MS780-E/H currently under test
                ;3141 ;
                ;3142 ; Outputs:
                ;3143 ; -------
                ;3144 ;
                ;3145 ; NO specific outputs.
                ;3146 ; Above mentioned bits will be cleared.
                ;3147 ;
                ;3148 ; Side Effects:
                ;3149 ; ------------
                ;3150 ;
                ;3151 ; The contents of the following registers will be lost
                ;3152 ; Q, D, SC, VA
                ;3153 ;
                ;3154 ;
                ;3155 ;--
                ;3156 ;*****************************************************
                ;3157 MS780E.CLEANUP:
U 12B5, 0010,0038,0180,0970,0200,12B6  ;3158  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
                ;3159  D_0,      ; Get data to clear Read/Write bits
U 12B6, 0F00,003C,0180,0803,0000,12B7  ;3160   VA_VA+4    ; Point to CNFG Reg B
U 12B7, 0000,003C,0180,A800,0000,12B8  ;3161  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 12B8, 0818,0038,7980,0800,0000,12B9  ;3162  D_K[.30]    ; Prepare to form x30000780 to clear
                ;3163       ; ...CNFG Reg's C and D
U 12B9, 0B00,003C,0180,0800,0000,12BA  ;3164  D_D.SWAP    ; D = x30000000
                ;3165  Q_D,      ; Save in the Q Reg
U 12BA, 0818,0038,CDE0,0800,0000,12BB  ;3166   D_K[.F0]   ; D = xF0
U 12BB, 0000,003C,0D80,0800,0084,72BC  ;3167  SC_K[.3]    ; Must shift D Reg left 3 bits to get
                ;3168       ; ...x780
```

```
U 12BC, 0D00,003C,0180,0800,0000,12BD  ;3169  D_DAL.SC    ; D = x780
          ;3170  D_D.OR.Q,    ; D = x30000780
U 12BD, 081D,0030,0180,0803,0000,12BE  ;3171   VA_VA+4    ; Point to CNFG Reg C
U 12BE, 0000,003C,0180,A800,0000,12BF  ;3172  CACHE.P_D[LONG]    ; Clear Bits in CNFG Reg C
U 12BF, 0000,003C,0180,0800,0000,12C0  ;3173  NOP
U 12C0, 0000,003C,0180,A800,0000,12C1  ;3174  CACHE.P_D[LONG]    ; Do second write to make sure that
          ;3175       ; ...uSequencer Parity error bit
          ;3176       ; ...is clear
U 12C1, 0000,003C,0180,0803,0000,12C2  ;3177  VA_VA+4     ; Point to CNFG Reg D
U 12C2, 0000,003C,0180,A800,0000,12C3  ;3178  CACHE.P_D[LONG]    ; Clear bits in CNFG Reg D
U 12C3, 0000,003C,0180,0800,0000,12C4  ;3179  NOP
          ;3180  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
          ;3181       ; ...Parity error bit
U 12C4, 0000,003E,0180,A800,0000,0001  ;3182   RETURN1    ; Return to caller
          ;3183
```

```
            ;3184  .PAGE
            ;3185  .TOC "   CHECK SBI.ERR"
            ;3186 ;**
            ;3187 ; FUNCTIONAL DESCRIPTION:
            ;3188 ;
            ;3189 ; This subroutine can be used to see whether the SBI.ERR Register
            ;3190 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;3191 ; routine will have to pass to this subroutine in the SC register
            ;3192 ; the number of the error bit to checked.
            ;3193 ;
            ;3194 ; CALLING CONSTRAINT:
            ;3195 ;
            ;3196 ; =00
            ;3197 ;
            ;3198 ; INPUTS:
            ;3199 ;
            ;3200 ; SC with the number of the error bit to be checked
            ;3201 ;
            ;3202 ; OUTPUTS:
            ;3203 ;
            ;3204 ; RC[08]-Contains the SBI.ERROR Register
            ;3205 ;--
            ;3206 CHECK_SBI_ERR:
U 12C5, 0000,003C,65F0,2C00,0000,12C6  ;3207  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 12C6, 0001,203C,0180,09C0,0000,12C7  ;3208  RC[08]_Q  ; Save for error logout
            ;3209  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 12C7, 0001,2024,0180,0800,0010,12C8  ;3210   CLK.UBCC ; ...
U 12C8, 0000,013C,0180,0800,0000,10D0  ;3211  Z?   ; Check condition codes
U 10D0, 0000,003E,0180,0800,0000,0002  ;3212 =0 RETURN2   ; Bit is set
U 10D1, 0000,003E,0180,0800,0000,0001  ;3213  RETURN1   ; Bit is not set
            ;3214
```

```
            ;3215  .PAGE
            ;3216  .TOC "    64K OR 256K CHIPS"
            ;3217 ;*************************************************************
            ;3218 ;**
            ;3219 ;
            ;3220 ; Functional Description:
            ;3221 ; ----------------------
            ;3222 ; This subroutine is used to find the type of chips
            ;3223 ; on the MS780-E/H arrays.  The RETURN modes are as
            ;3224 ; follows:
            ;3225 ;
            ;3226 ; RETURN1 = Error. Mixed 64K and 256K arrays
            ;3227 ;
            ;3228 ; RETURN2 = 64K chips on the array
            ;3229 ;
            ;3230 ; RETURN3 = 256K chips on the array
            ;3231 ;
            ;3232 ; Calling Constraint: =00
            ;3233 ; ------------------
            ;3234 ;
            ;3235 ;
            ;3236 ; Inputs:
            ;3237 ; ------
            ;3238 ;
            ;3239 ; RC[0E] must hold the I/O base address of the MS780-E/H
            ;3240 ; currently under test
            ;3241 ;
            ;3242 ; Outputs:
            ;3243 ; -------
            ;3244 ;
            ;3245 ; NO specific outputs. (See RETURN modes above)
            ;3246 ;
            ;3247 ;
            ;3248 ; Side Effects:
            ;3249 ; ------------
            ;3250 ;
            ;3251 ; The contents of the following registers will be lost:
            ;3252 ;
            ;3253 ; D
            ;3254 ;
            ;3255 ;--
            ;3256 ;*************************************************************
            ;3257 64K.OR.256K:
U 12C9, 0010,0038,0180,0970,0200,12CA  ;3258  VA_RC[0E]    ; Point to CNFG Register A
U 12CA, 0000,003C,0180,C800,0000,12CB  ;3259  D[LONG]_CACHE.P    ; Read the register
U 12CB, 0200,003C,0180,0800,0000,12CC  ;3260  D_D.RIGHT2    ; Shift received contents two bits
            ;3261       ; ...to the right
U 12CC, 0600,003C,0180,0800,0000,12CD  ;3262  D_D.RIGHT    ; Shift D one more time
U 12CD, 0000,193C,0180,0800,0000,102C  ;3263  D1-0?    ; Branch on the RAM Type
U 102C, 0000,003E,0180,0800,0000,0001  ;3264 =1100 RETURN1    ; Error: both 64K and 256K RAMs present
U 102D, 0000,003E,0180,0800,0000,0002  ;3265  RETURN2    ; 64K RAMs found
U 102E, 0000,003E,0180,0800,0000,0003  ;3266  RETURN3    ; 256K RAMs found
U 102F, 0000,003E,0180,0800,0000,0001  ;3267  RETURN1    ; Error: missing arrays
            ;3268
            ;3269
```

```
            ;3270  .PAGE
            ;3271  .TOC "   GET MEMORY SIZE"
            ;3272 ;*************************************************************
            ;3273 ;**
            ;3274 ;
            ;3275 ; Functional Description:
            ;3276 ; ----------------------
            ;3277 ;
            ;3278 ; This subroutine can be used to get the size of the memory.
            ;3279 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3280 ; increments.
            ;3281 ;
            ;3282 ; Calling Constraint: =0
            ;3283 ; ------------------
            ;3284 ;
            ;3285 ;
            ;3286 ; Inputs:
            ;3287 ; ------
            ;3288 ;
            ;3289 ; RC[0E] Must have the Address of CNFG Register A
            ;3290 ;
            ;3291 ; Outputs:
            ;3292 ; -------
            ;3293 ;
            ;3294 ; Register D will contain upon exit the size of the
            ;3295 ; memory aligned on Mega-byte boundary.
            ;3296 ;
            ;3297 ; Side Effects:
            ;3298 ; ------------
            ;3299 ;
            ;3300 ; The contents of the following registers will be lost:
            ;3301 ;
            ;3302 ; D, SC, Q
            ;3303 ;
            ;3304 ;--
            ;3305 ;*************************************************************
            ;3306 GET.MEMORY.SIZE:
U 12CE, 0010,0038,0180,0970,0200,12CF  ;3307  VA_RC[0E]    ; Point to CNFG Reg A
U 12CF, 0000,003C,0180,C800,0000,12D0  ;3308  D[LONG]_CACHE.P    ; Read the register
U 12D0, 001B,0000,D980,0800,0082,12D1  ;3309  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 12D1, 0D00,003C,0180,0800,0000,12D2  ;3310  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 12D2, 0819,0034,5580,0800,0000,1030  ;3311  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3312 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1030, 0000,003D,01E0,0800,0000,12C9  ;3313   Q_D    ; Save Memory size bits in Q
U 1031, 0000,003D,0D80,0800,0084,709C  ;3314  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3315       ; ...arrays on the controller
            ;3316 ;
            ;3317 ; The duplication of the next two return status is necessary so the returning
            ;3318 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3319 ; modifications.
            ;3320 ;
U 1032, 0819,2014,0580,0800,0000,1033  ;3321  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3322       ; ...(found 1 Meg arrays)
U 1033, 0819,2014,0580,0800,0000,12D3  ;3323  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3324       ; ...(found 4 Meg arrays)
```

```
U 12D3, 0000,003C,21F8,0800,0084,72D4  ;3325  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
        ;3326  D_DAL.SC,   ; Shift bits <5:0> to <25:20>
U 12D4, 0D00,003E,0180,0800,0000,0001  ;3327   RETURN1   ; Return with memory size in Register D
        ;3328
```

```
;3329  .PAGE
;3330  .TOC "   FORCE CRD-RDS ERROR AND READ"
;3331 ;*************************************************************
;3332 ;**
;3333 ;
;3334 ; Functional Description:
;3335 ; ----------------------
;3336 ;
;3337 ;
;3338 ;      This subroutine will be used to force any combination of CRD
;3339 ; or RDS errors (or no errors whatsoever) in locations 0 and 4.  It
;3340 ; should be passed from the calling routine in Reg D  the  data  to
;3341 ; force  an  error  in location 0 and in Reg Q the data to force an
;3342 ; error in location 4.  The subroutine will  initialize  the  array
;3343 ; locations 0 and 4 with the data in registers D and Q, will select
;3344 ; Diagnostic Mode 2 and set the ECC substitute bits in CNFG  Reg  B
;3345 ; to  x0C.   An Extended Read will follow, thus forcing the errors,
;3346 ; followed by a Return1.
;3347 ;
;3348 ;
;3349 ; Calling Constraint: =0
;3350 ; ------------------
;3351 ;
;3352 ;
;3353 ; Inputs:
;3354 ; ------
;3355 ;
;3356 ; Registers D and Q MUST hold the following information:
;3357 ;
;3358 ; D - Data to force error in location 0
;3359 ;
;3360 ; Q - Data to force error in location 4
;3361 ;
;3362 ;
;3363 ; Outputs:
;3364 ; -------
;3365 ;
;3366 ; There are NO specific outputs from this routine.
;3367 ; However depending on the data in the D and Q registers
;3368 ; any combination of CRD/RDS errors  (or NO error at all)
;3369 ; will be forced in the memory.
;3370 ;
;3371 ;
;3372 ; Side Effects:
;3373 ; ------------
;3374 ;
;3375 ; The contents of the following registers will be lost:
;3376 ;
;3377 ; D, Q, VA, SC
;3378 ;
;3379 ;--
;3380 ;*************************************************************
;3381 FORCE.CRD.RDS.READ:
U 12D5, 0018,0038,1980,0800,0200,12D6  ;3382  VA_K[ZERO]    ; Point VA to location 0
U 12D6, 0000,003C,0180,A800,0000,12D7  ;3383  CACHE.P_D[LONG]    ; Intialize location 0 with the data
```

```
            ;3384  D_Q,     ; Get data for location 4
U 12D7, 0C00.003C.0180.0803.0000.12D8  ;3385    VA_VA+4    ; Point to location 4
U 12D8, 0000.003C.0180.A800.0000.10D2  ;3386   CACHE.P_D[LONG]    ; Initialize location 4
U 10D2, 0000.003D.8580.0800.0084.726B  ;3387 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 10D3, 0000.003C.0180.0800.0000.10D4  ;3388   NOP
U 10D4, 0818.0039.0980.0800.0000.1243  ;3389 =0 SET.DIAG.MODE[.2]  ; Set ECC substitute Diagnostic Mode
U 10D5, 0000.003C.0180.0800.0000.10D6  ;3390   NOP
U 10D6, 0818.0039.8580.0800.0000.12AF  ;3391 =0 SET.ECC[.C]    ; Set ECC substitute bits for data of
            ;3392          ; ...zeroes
U 10D7, 0018.0038.1980.0800.0200.12D9  ;3393   VA_K[ZERO]    ; Point VA to location 0
            ;3394  D[LONG]_CACHE.P,   ; Read location 0 forcing the ECC errors
U 12D9, 0000.003E.0180.C800.0000.0001  ;3395    RETURN1    ; Return to caller
            ;3396
```

```
;3397  .PAGE "TEST 1C7 CRD INHIBIT"
;3398 ;*************************************************************
;3399 ;***
;3400 ;
;3401 ; Functional Description:
;3402 ; ----------------------
;3403 ;
;3404 ; This test verifies that bit 30, CRD inhibit, of CNFG C (or D) register
;3405 ; functions correctly.  A CRD error is forced while this bit is set, and
;3406 ; it is expected that the CPU will not detect the error (ie., no CRD tag
;3407 ; was sent to the CPU on the SBI).
;3408 ;
;3409 ;
;3410 ; Logic Description:
;3411 ; -----------------
;3412 ;  N/A
;3413 ;
;3414 ; Error Logout:
;3415 ; ------------
;3416 ;
;3417 ; See individual error reports.
;3418 ;
;3419 ; Sync Point Description:
;3420 ; ----------------------
;3421 ;  N/A
;3422 ;
;3423 ;=============================================================
;3424 ;
;3425 ; Register Map:
;3426 ; ------------
;3427 ;
;3428 ;      RA,RB      Description:
;3429 ;      -----      -----------
;3430 ;
;3431 ; Not Used
;3432 ;
;3433 ;       RC        Description:
;3434 ;       --        -----------
;3435 ;
;3436 ;        E I/O Base Address of MS780-E Currently Under Test
;3437 ;
;3438 ; D I/O Address of Configuration Register C/D
;3439 ;
;3440 ; C Contents of Configuration Register C/D
;3441 ;
;3442 ;--
;3443 ;*************************************************************
;3444 =0
U 10D8, 0000,003D,0180,0800,0000,1093  ;3445 T.39: NEWTST   ; Signify the start of a new test
U 10D9, 0000,003C,0180,0800,0000,1034  ;3446  NOP
U 1034, 0000,003D,0180,0800,0000,10B2  ;3447 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the system
U 1035, 0000,003C,0180,0800,0000,10F8  ;3448  J/T.39.EXIT  ; No MS780-E's on the SBI to test, exit
U 1036, 0000,003C,0180,0800,0000,1038  ;3449  NOP
          ;3450 =
          ;3451 =00
```

```
            ;3452  ;
            ;3453  ; This is the point for restarting the test to check the next
            ;3454  ; controller.
            ;3455  ;
            ;3456 RESTART.TEST.39:
U 1038. 0000.003D.0180.0800.0000.10A2  ;3457  CALL[START.TEST] ; Configure the Controllers
U 1039. 0000.003C.0180.0800.0000.10F8  ;3458  J/T.39.EXIT  ; No more controllers to configure, exit test
U 103A. 0000.003D.0180.0800.0000.1228  ;3459  ERLOOP   ; Loop on error from here
U 103B. 0000.003C.0180.0800.0000.10DA  ;3460  NOP
U 10DA. 0000.003D.0180.0800.0000.10AC  ;3461 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10DB. 0000.003C.0180.0800.0000.10DC  ;3462  NOP
U 10DC. 0000.003D.0180.0800.0000.12B5  ;3463 =0 CALL[MS780E.CLEANUP] ; Clean up the Configuration Registers
U 10DD. 0010.0038.0180.0968.0200.12DA  ;3464  VA_RC[0D]  ; Point VA to configuration register C/D
U 12DA. 0000.003C.0180.C800.0000.12DB  ;3465  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 12DB. 0001.003C.0180.09D0.0000.12DC  ;3466  RC[0A]_D  ; Save in RC[0A]
U 12DC. 0000.003C.5180.0800.0084.72DD  ;3467  SC_K[.1E]  ; Set up to mask bit 30 in CNFG C/D
U 12DD. 0003.003C.0180.09E0.0000.12DE  ;3468  RC[0C]_0  ; Expected status of the bit
            ;3469  ALU_D.ANDNOT.MASK, ; Mask out bit 30 ( Inhibit CRD ) in
            ;3470   RC[0B]_ALU,
U 12DE. 0001.0024.0180.09D8.0010.12DF  ;3471   CLK.UBCC ; Configuration Register C/D
U 12DF. 0000.013C.0180.0800.0000.10DE  ;3472  Z?   ; Was bit 30 set ?
U 10DE. 0000.003C.0180.0800.0000.10E0  ;3473 =0 J/T.39.BIT30.SET.1 ; Yes, shouldn't have been, Call an error
U 10DF. 0000.003C.0180.0800.0000.10E1  ;3474  J/T.39.BIT30.CLEAR.1 ; No, continue with test
```

```
            ;3475  .PAGE
            ;3476 ;
            ;3477 ;////////////////////////////////////////////////////////////////////////////////
            ;3478 ;
            ;3479 ; Bit 30 ( Inhibit CRD ) in Configuration Register C/D was not clear and should
            ;3480 ; have been.  Call an error.
            ;3481 ;
            ;3482 =0
            ;3483 T.39.BIT30.SET.1:
U 10E0, 0000.003D.D580.0800.0084.709E  ;3484  ERROR2[.6]
            ;3485 ;
            ;3486 ;////////////////////////////////////////////////////////////////////////////////
            ;3487 ;
            ;3488 ; ERROR LOGOUT:
            ;3489 ;
            ;3490 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3491 ;  I/O Address of Cnfg Reg C/D of controller under test
            ;3492 ;  Expected status of bit 30 in Cnfg Reg C/D
            ;3493 ;  Received status of bit 30 in Cnfg Reg C/D
            ;3494 ;  Contents of Configuration Register C/D
            ;3495 ;  Not Used
            ;3496 ;
            ;3497 ;////////////////////////////////////////////////////////////////////////////////
            ;3498 ;
            ;3499 T.39.BIT30.CLEAR.1:
U 10E1, 0010.0038.0180.0968.0200.12E0  ;3500  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 12E0, 0000.003C.5180.0800.0084.72E1  ;3501  SC_K[.1E]  ; Get ready to try setting bit 30 in CNFG C/D
U 12E1, 0803.001C.0180.0800.0000.12E2  ;3502  D_NOT.MASK  ; Load D register with x40000000
U 12E2, 0000.003C.0180.A800.0000.12E3  ;3503  CACHE.P_D[LONG]  ; Try to set bit 30 in CNFG C/D
U 12E3, 0000.003C.0180.0800.0000.12E4  ;3504  NOP  ; ...
U 12E4, 0000.003C.0180.C800.0000.12E5  ;3505  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 12E5, 0001.003C.0180.09D0.0000.12E6  ;3506  RC[0A]_D  ; Save in RC[0A]
U 12E6, 0000.003C.5180.0800.0084.72E7  ;3507  SC_K[.1E]  ; Set up to mask bit 30 in CNFG C/D
            ;3508  ALU_NOT.MASK,  ; Set expected value for bit 30 in CNFG Reg C/D
U 12E7, 0003.001C.0180.09E0.0000.12E8  ;3509   RC[0C]_ALU ; ...
            ;3510  ALU_D.ANDNOT.MASK, ; Mask out bit 30 ( Inhibit CRD ) in CNFG C/D
            ;3511   RC[0B]_ALU, ; ...and save for error logout
U 12E8, 0001.0024.0180.09D8.0010.12E9  ;3512   CLK.UBCC ; ...
U 12E9, 0000.013C.0180.0800.0000.10E2  ;3513  Z?  ; Was bit 30 set ?
U 10E2, 0000.003C.0180.0800.0000.10E6  ;3514 =0 J/T.39.BIT30.SET.2 ; Yes, continue with test
U 10E3, 0000.003C.0180.0800.0000.10E4  ;3515  NOP  ; No, should have been, call an error
```

```
            ;3516  .PAGE
            ;3517 ;
            ;3518 ;////////////////////////////////////////////////////////////////////////////
            ;3519 ;
            ;3520 ; Bit 30 ( Inhibit CRD ) in Configuration Register C/D was not set and should
            ;3521 ; have been.  Call an error.
            ;3522 ;
            ;3523 =0
U 10E4, 0000,003D,D580,0800,0084,709E  ;3524  ERROR2[.6]
U 10E5, 0000,003C,0180,0800,0000,10E6  ;3525  NOP
            ;3526 ;
            ;3527 ;////////////////////////////////////////////////////////////////////////////
            ;3528 ;
            ;3529 ; ERROR LOGOUT:
            ;3530 ;
            ;3531 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3532 ;  I/O Address of Cnfg Reg C/D of controller under test
            ;3533 ;  Expected status of bit 30 in Cnfg Reg C/D
            ;3534 ;  Received Status of bit 30 in Cnfg Reg C/D
            ;3535 ;  Contents of Configuration Register C/D
            ;3536 ;  Not Used
            ;3537 ;
            ;3538 ;////////////////////////////////////////////////////////////////////////////
            ;3539 ;
            ;3540 =0
            ;3541 T.39.BIT30.SET.2:
U 10E6, 0000,003D,0180,0800,0000,1228  ;3542  ERLOOP   ; Loop on error from here
U 10E7, 0000,003C,0180,0800,0000,10E8  ;3543  NOP
U 10E8, 0000,003D,0180,0800,0000,10AC  ;3544 =0 CALL[SBI.INIT]  ; Initialize the SBI
            ;3545  D_K[.1],  ; Set up to force a CRD Error
U 10E9, 0818,0038,05F8,0800,0000,10EA  ;3546   Q_0  ; Clear the Q register
U 10EA, 0000,003D,0180,0800,0000,12D5  ;3547 =0 CALL[FORCE.CRD.RDS.READ]; Force the CRD error
U 10EB, 0818,0038,0180,0800,0000,10EC  ;3548  D_K[.8]   ; Get number of cycles to wait for SBI to
            ;3549      ; finish the cycle
U 10EC, 0000,003D,0180,0800,0000,1257  ;3550 =0 CALL[WAIT.LOOP]  ; Do the wait
U 10ED, 001B,0010,8980,0800,0082,103C  ;3551  SC_K[.D]+1  ; Get ready to check bit 14 in SBI Error reg
U 103C, 0000,003D,0180,0800,0000,12C5  ;3552 =00 CALL[CHECK_SBI_ERR] ; Was bit 14 set ?
U 103D, 0000,003C,0180,0800,0000,103F  ;3553  J/T.39.BIT14.NOTSET ; No, continue with the test
```

```
            ;3554  .PAGE
            ;3555 ;
            ;3556 ;////////////////////////////////////////////////////////////////////////////
            ;3557 ;
            ;3558 ; SBI Error register bit 14 ( CRD ) was set and shouldn't have been.
            ;3559 ; Call an error.
            ;3560 ;
U 103E, 0000,003D,5D80,0800,0084,709E  ;3561  ERROR2[.7]
            ;3562 ;
            ;3563 ;////////////////////////////////////////////////////////////////////////////
            ;3564 ;
            ;3565 ; ERROR LOGOUT:
            ;3566 ;
            ;3567 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3568 ;  I/O Address of Configuration Register C/D
            ;3569 ;  Not Used
            ;3570 ;  Not Used
            ;3571 ;  Not Used
            ;3572 ;  Not Used
            ;3573 ;  ID[sbi.err] Register
            ;3574 ;
            ;3575 ;////////////////////////////////////////////////////////////////////////////
            ;3576 ;
            ;3577 T.39.BIT14.NOTSET:
U 103F, 0010,0038,0180,0968,0200,12EA  ;3578  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 12EA, 0000,003C,0180,C800,0000,12EB  ;3579  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 12EB, 0001,003C,0180,09D0,0000,12EC  ;3580  RC[0A]_D  ; Save in RC[0A]
U 12EC, 0000,003C,D980,0800,0084,72ED  ;3581  SC_K[.9]  ; Set up to mask bit 9 in CNFG C/D
            ;3582  ALU_NOT.MASK,  ; Set Expected Status of bit 9 in Cnfg Reg C/D
U 12ED, 0003,001C,0180,09E0,0000,12EE  ;3583  RC[0C]_ALU ; ...for error logout
            ;3584  ALU_D.ANDNOT.MASK, ; Mask out bit 9 ( CRD Flag ) in
            ;3585  RC[0B]_ALU, ; Configuration Register C/D
U 12EE, 0001,0024,0180,09D8,0010,12EF  ;3586  CLK.UBCC ; ...and save for error logout
U 12EF, 0000,013C,0180,0800,0000,10EE  ;3587  Z?  ; Was bit 9 set ?
U 10EE, 0000,003C,0180,0800,0000,10F2  ;3588 =0 J/T.39.BIT9.SET  ; OK. Bit is set
U 10EF, 0000,003C,0180,0800,0000,10F0  ;3589  J/T.39.BIT9.CLEAR ; No, Call an error
            ;3590
```

```
            ;3591   .PAGE
            ;3592 ;
            ;3593 ;////////////////////////////////////////////////////////////////////////////////
            ;3594 ;
            ;3595 ; Bit 9 ( CRD Flag ) in Configuration Register C/D was NOT set
            ;3596 ; Call an error.
            ;3597 ;
            ;3598 =0
            ;3599 T.39.BIT9.CLEAR:
U 10F0, 0000.003D.D580.0800.0084.709E  ;3600  ERROR2[.6]
U 10F1, 0000.003C.0180.0800.0000.10F2  ;3601  NOP
            ;3602 ;
            ;3603 ;////////////////////////////////////////////////////////////////////////////////
            ;3604 ;
            ;3605 ; ERROR LOGOUT:
            ;3606 ;
            ;3607 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3608 ;  I/O Address of Cnfg Reg C/D of controller under test
            ;3609 ;  Expected Status of bit 9 in Cnfg Reg C/D
            ;3610 ;  Received Status of bit 9 in Cnfg Reg C/D
            ;3611 ;  Contents of Configuration Register C/D
            ;3612 ;  Not Used
            ;3613 ;
            ;3614 ;////////////////////////////////////////////////////////////////////////////////
            ;3615 ;
            ;3616 =0
            ;3617 T.39.BIT9.SET:
U 10F2, 0000.003D.0180.0800.0000.1228  ;3618  ERLOOP   ; Loop on error from here
U 10F3, 0000.003C.0180.0800.0000.10F4  ;3619  NOP
U 10F4, 0000.003D.0180.0800.0000.10AC  ;3620 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10F5, 0F00.003C.0180.0800.0000.12F0  ;3621  D_0   ; Get ready to write zero to CNFG C/D
U 12F0, 0000.003C.0180.A800.0000.12F1  ;3622  CACHE.P_D[LONG]  ; Do write to try clearing CRD inhibit Bit
U 12F1, 0000.003C.0180.0800.0000.12F2  ;3623  NOP
U 12F2, 0000.003C.0180.C800.0000.12F3  ;3624  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 12F3, 0001.003C.0180.09D0.0000.12F4  ;3625  RC[0A]_D  ; Save in RC[0A]
U 12F4, 0000.003C.5180.0800.0084.72F5  ;3626  SC_K[.1E]  ; Set up to mask bit 30 in CNFG C/D
U 12F5, 0003.003C.0180.09E0.0000.12F6  ;3627  RC[0C]_0  ; Set expected Status of bit 30 in Cnfg Reg C/D
            ;3628     ; ...for error logout
            ;3629  ALU_D.ANDNOT.MASK, ; Mask out bit 30 ( Inhibit CRD ) in
            ;3630   RC[0B]_ALU, ; ...Configuration Register C/D
U 12F6, 0001.0024.0180.09D8.0010.12F7  ;3631   CLK.UBCC ; ...and save for error logout
U 12F7, 0000.013C.0180.0800.0000.10F6  ;3632  Z?   ; Was bit 30 set ?
            ;3633     ; the next MS780-E controller
```

```
            ;3634  .PAGE
            ;3635 ;
            ;3636 ;////////////////////////////////////////////////////////////////////////////
            ;3637 ;
            ;3638 ; Bit 30 ( Inhibit CRD ) in Configuration Register C/D was set and should
            ;3639 ; NOT have been.  Call an error.
            ;3640 ;
            ;3641 =0
            ;3642 T.39.BIT30.SET.3:
U 10F6, 0000,003D,D580,0800,0084,709E  ;3643  ERROR2[.6]  ; I.E. bit 30 did not clear
U 10F7, 0000,003C,0180,0800,0000,1038  ;3644  J/RESTART.TEST.39
            ;3645 ;
            ;3646 ;////////////////////////////////////////////////////////////////////////////
            ;3647 ;
            ;3648 ; ERROR LOGOUT:
            ;3649 ;
            ;3650 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3651 ;  I/O Address of Cnfg Reg C/D of controller under test
            ;3652 ;  Expected Status of bit 30 in Cnfg Reg C/D
            ;3653 ;  Received Status of bit 30 in Cnfg Reg C/D
            ;3654 ;  Contents of Cnfg Reg C/D
            ;3655 ;  Not Used
            ;3656 ;
            ;3657 ;////////////////////////////////////////////////////////////////////////////
            ;3658 ;
            ;3659  ;
            ;3660  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;3661  ;
            ;3662 =0
            ;3663 T.39.EXIT:
U 10F8, 0000,003D,0180,0800,0000,12B5  ;3664  CALL[MS780E.CLEANUP]  ; Clear Memory CNFG Regs before
U 10F9, 0000,003C,0180,0800,0000,10FA  ;3665  NOP     ; ...exitting
            ;3666
            ;3667
```

```
;3668   .PAGE "TEST 1C8 ERROR LOG AND HIGH ERROR RATE BITS TEST"
;3669  ;*************************************************************
;3670  ;***
;3671  ;
;3672  ; Functional Description:
;3673  ; ----------------------
;3674  ;
;3675  ; The test verifies that the High Error Rate bit, bit 29 of CNFG
;3676  ; Register C/D, and the Error Log Bit, bit 28 in the same register,
;3677  ; set and clear properly.
;3678  ;
;3679  ; The High Error Rate bit should only set when two consecutive errors
;3680  ; take place and the error bits are not cleared between them.  If a
;3681  ; single error occurs it is expected that the bit will no set.
;3682  ;
;3683  ; SUBTEST 1
;3684  ;
;3685  ; TWO CRD ERRORS
;3686  ;
;3687  ; SUBTEST 2
;3688  ;
;3689  ; TWO RDS ERRORS
;3690  ;
;3691  ; SUBTEST 3
;3692  ;
;3693  ; CRD FOLLOWED BY RDS
;3694  ;
;3695  ; SUBTEST 4
;3696  ;
;3697  ; RDS FOLLOWED BY CRD
;3698  ;
;3699  ; Logic Description:
;3700  ; -----------------
;3701  ; N/A
;3702  ;
;3703  ; Error Logout:
;3704  ; ------------
;3705  ;
;3706  ; See individual error reports.
;3707  ;
;3708  ; Sync Point Description:
;3709  ; ----------------------
;3710  ; N/A
;3711  ;
;3712  ;=============================================================
;3713  ;
;3714  ; Register Map:
;3715  ; ------------
;3716  ;
;3717  ;     RA,RB     Description:
;3718  ;     -----     -----------
;3719  ;
;3720  ;       0       Data to force first error
;3721  ;
;3722  ;       1 Data to force second error
```

```
            ;3723 ;
            ;3724 ;
            ;3725 ;       RC       Description:
            ;3726 ;       --       -----------
            ;3727 ;
            ;3728 ;        E I/O Base Address of MS780-E Currently Under Test
            ;3729 ;
            ;3730 ;        D I/O Address of Configuration Register C/D
            ;3731 ;
            ;3732 ;
            ;3733 ;--
            ;3734 ;*************************************************************
            ;3735 =0
U 10FA. 0000,003D,0180,0800,0000,1093  ;3736 T.40: NEWTST   ; Signify the start of a new test
U 10FB. 0000,003C,0180,0800,0000,1040  ;3737  NOP
U 1040. 0000,003D,0180,0800,0000,10B2  ;3738 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1041. 0000,003C,0180,0800,0000,1144  ;3739  J/T.40.EXIT  ; No MS780-E's on the SBI to test, exit test
U 1042. 0000,003C,0180,0800,0000,1044  ;3740  NOP
            ;3741 =
            ;3742  ;
            ;3743  ; This entry point is where the test is restarted to check the
            ;3744  ; next MS780-E controller
            ;3745  ;
            ;3746 =00
            ;3747 RESTART.TEST.40:
U 1044. 0000,003D,0180,0800,0000,10A2  ;3748  CALL[START.TEST] ; Configure the Controllers
U 1045. 0000,003C,0180,0800,0000,1144  ;3749  J/T.40.EXIT  ; No more controllers to configure, exit
U 1046. 0000,003D,0180,0800,0000,1228  ;3750  ERLOOP   ; Loop on error from here
U 1047. 0000,003C,0180,0800,0000,10FC  ;3751  NOP
U 10FC. 0000,003D,0180,0800,0000,10AC  ;3752 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10FD. 0000,003C,0180,0800,0000,10FE  ;3753  NOP
U 10FE. 0000,003D,0180,0800,0000,12B5  ;3754 =0 CALL[MS780E.CLEANUP] ; Clear out the Configuration Registers
U 10FF. 0010,0038,0180,0968,0200,12F8  ;3755  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 12F8. 0000,003C,0180,C800,0000,12F9  ;3756  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 12F9. 0001,003C,0180,09D0,0000,12FA  ;3757  RC[0A]_D  ; Save read data in RC[0A]
U 12FA. 0003,003C,0180,09E0,0000,12FB  ;3758  RC[0C]_0  ; Set Expected status of bits 28 and 29
            ;3759      ; ...in Cnfg Reg C/D
U 12FB. 001B,0010,ED80,0800,0082,12FC  ;3760  SC_K[.1B]+1  ; Get ready to mask bit 28
U 12FC. 0803,001C,0180,0800,0000,12FD  ;3761  D_NOT.MASK  ; Assert bit 28 in the D register and
            ;3762  Q_D,   ; Save in Q
U 12FD. 0000,003C,01E0,0800,0080,D2FE  ;3763  SC_SC+1   ; Get ready to assert bit 29
U 12FE. 0803,001C,0180,0800,0000,12FF  ;3764  D_NOT.MASK  ; Assert bit 29 in the D register
U 12FF. 081D,0030,0180,0800,0000,1300  ;3765  D_D.OR.Q  ; Now Or D with Q so bit 28 & 29 are set
            ;3766  Q_D.AND.RC[0C],  ; Mask (bits 28:29 set) in
U 1300. 0011,0034,01C0,0960,0010,1301  ;3767   CLK.UBCC ; ...configuration register C/D data
U 1301. 0001,213C,0180,09D8,0000,110A  ;3768  Z?,RC[0B]_Q  ; Were bits 28 and 29 clear ?
U 110A. 0000,003C,0180,0800,0000,1110  ;3769 =0 J/T.40.28.29.NOT.CLEAR.1; No, should have been, call an error
U 110B. 0000,003C,0180,0800,0000,1111  ;3770  J/T.40.28.29.CLEAR.1 ; Yes, continue with the test
```

```
            ;3771   .PAGE
            ;3772 ;
            ;3773 ;////////////////////////////////////////////////////////////////////////////////
            ;3774 ;
            ;3775 ; Either of bits 28 and 29 were set in configuration Register C/D.
            ;3776 ; Call an error.
            ;3777 ;
            ;3778 =0
            ;3779 T.40.28.29.NOT.CLEAR.1:
U 1110, 0000,003D,D580,0800,0084,709E  ;3780  ERROR2[.6]
            ;3781 ;
            ;3782 ;////////////////////////////////////////////////////////////////////////////////
            ;3783 ;
            ;3784 ; ERROR LOGOUT:
            ;3785 ;
            ;3786 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3787 ;  I/O Address of Cnfg Reg C/D of Controller under test
            ;3788 ;  Expected Status of bits 28 and 29
            ;3789 ;  Received Status of bits 28 and 29
            ;3790 ;  Contents of Configuration Register C/D
            ;3791 ;  Not Used
            ;3792 ;
            ;3793 ;////////////////////////////////////////////////////////////////////////////////
            ;3794 ;
            ;3795 T.40.28.29.CLEAR.1:
            ;3796 ;
            ;3797 ;================================================================================
            ;3798 ;
            ;3799 ; Subtest 1
            ;3800 ;
U 1111, 0018,0038,0580,0A80,0000,1302  ;3801  R[0]_K[.1]  ; First error will be a CRD
U 1302, 0018,0038,0580,0A88,0000,1112  ;3802  R[1]_K[.1]  ; Second Error will be a CRD
U 1112, 0000,003D,0180,0800,0000,111A  ;3803 =0 CALL[TEST.40.TST] ; Execute the test
            ;3804 ;
            ;3805 ;================================================================================
            ;3806 ;
            ;3807 ; Subtest 2
            ;3808 ;
U 1113, 0018,0038,0D80,0A80,0000,1303  ;3809  R[0]_K[.3]  ; First error will be a RDS
U 1303, 0018,0038,0D80,0A88,0000,1114  ;3810  R[1]_K[.3]  ; Second error will be a RDS
U 1114, 0000,003D,0180,0800,0000,111A  ;3811 =0 CALL[TEST.40.TST] ; Execute the test
            ;3812 ;
            ;3813 ;================================================================================
            ;3814 ;
            ;3815 ; Subtest 3
            ;3816 ;
U 1115, 0018,0038,0580,0A80,0000,1304  ;3817  R[0]_K[.1]  ; First error will be a CRD
U 1304, 0018,0038,0D80,0A88,0000,1116  ;3818  R[1]_K[.3]  ; Second error will be a RDS
U 1116, 0000,003D,0180,0800,0000,111A  ;3819 =0 CALL[TEST.40.TST] ; Execute the test
            ;3820 ;
            ;3821 ;================================================================================
            ;3822 ;
            ;3823 ; Subtest 4
            ;3824 ;
U 1117, 0018,0038,0D80,0A80,0000,1305  ;3825  R[0]_K[.3]  ; First error will be a RDS
```

```
U 1305, 0018.0038.0580.0A88.0000.1118  ;3826  R[1]_K[.1]  ; Second error will be a CRD
U 1118, 0000.003D.0180.0800.0000.111A  ;3827 =0 CALL[TEST.40.TST] ; Execute the test
U 1119, 0000.003C.0180.0800.0000.1044  ;3828  J/RESTART.TEST.40 ; Restart this test for the next controller
          ;3829  ;
          ;3830  ; To decrease the amount of space used by this test the test itself
          ;3831  ; will be setup as a subroutine.  The parameters that should be
          ;3832  ; passed from the calling routine are the type of errors to be
          ;3833  ; forced.  The following code is the test subroutine.
          ;3834  ;
          ;3835 =0
          ;3836 TEST.40.TST:
U 111A, 0000.003D.0180.0800.0000.1230  ;3837  SUBTEST   ; Update the subtest counter
U 111B, 0000.003C.0180.0800.0000.111C  ;3838  NOP
U 111C, 0000.003D.0180.0800.0000.12B5  ;3839 =0 CALL[MS780E.CLEANUP] ; Clear all the error bits
U 111D, 0000.003C.0180.0800.0000.111E  ;3840  NOP
U 111E, 0000.003D.0180.0800.0000.1228  ;3841 =0 ERLOOP   ; Loop on error from here
U 111F, 0000.003C.0180.0800.0000.1120  ;3842  NOP
U 1120, 0000.003D.0180.0800.0000.10AC  ;3843 =0 CALL[SBI.INIT]  ; Initialize the SBI
          ;3844  D_R[0],   ; Get data for first error to force
U 1121, 0800.003C.01F8.0A00.0000.1122  ;3845   Q_0  ; Zero the Q register
U 1122, 0000.003D.0180.0800.0000.12D5  ;3846 =0 CALL[FORCE.CRD.RDS.READ]; Force the first error
U 1123, 0010.0038.0180.0968.0200.1306  ;3847  VA_RC[0D]  ; Point VA to configuration register c/d
U 1306, 0000.003C.0180.C800.0000.1307  ;3848  D[LONG]_CACHE.P  ; Read CNFG Register C/D
U 1307, 0001.003C.0180.09D0.0000.1308  ;3849  RC[0A]_D  ; Save CNFG C/D data in RC[0A]
U 1308, 0000.003C.5180.0800.0084.7309  ;3850  SC_K[.1E]  ; Get ready to mask bit 29
U 1309, 0000.003C.0580.0800.0084.B30A  ;3851  SC_SC-1   ; ...
U 130A, 0803.001C.0180.0800.0000.130B  ;3852  D_NOT.MASK  ; Assert bit 29 in the D register
U 130B, 0003.003C.0180.09E0.0000.130C  ;3853  RC[0C]_0  ; Save Expected Status of bit 29
          ;3854  Q_D.AND.RC[0A],   ; Mask bit 29
U 130C, 0011.C034.01C0.0950.0010.130D  ;3855   CLK.UBCC ; ...CNFG Register C/D data
U 130D, 0001.213C.0180.09D8.0000.1124  ;3856  Z?,RC[0B]_Q  ; Was bit 29 set ?
U 1124, 0000.003C.0180.0800.0000.1126  ;3857 =0 J/T.40.BIT29.SET.1 ; Yes, shouldn't have been, call an error
U 1125, 0000.003C.0180.0800.0000.1127  ;3858  J/T.40.BIT29.NOT.SET.1 ; No, continue with the test
```

```
            ;3859  .PAGE
            ;3860 ;
            ;3861 ;////////////////////////////////////////////////////////////////////////////
            ;3862 ;
            ;3863 ; Bit 29 ( High Error Rate ) in configuration register c/d was set and
            ;3864 ; shouldn't have been.  Call an error.
            ;3865 ;
            ;3866 =0
            ;3867 T.40.BIT29.SET.1:
U 1126. 0000,003D,D580,0800,0084,709E  ;3868  ERROR2[.6]
            ;3869 ;
            ;3870 ;////////////////////////////////////////////////////////////////////////////
            ;3871 ;
            ;3872 ; ERROR LOGOUT:
            ;3873 ;
            ;3874 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3875 ;  I/O Address of Cnfg  Reg C/D of Controller under test
            ;3876 ;  Expected Status of bit 29 in Cnfg Reg C/D
            ;3877 ;  Received Status of Bit 29 in Cnfg Reg C/D
            ;3878 ;  Contents of Configuration Register C/D
            ;3879 ;  Not Used
            ;3880 ;
            ;3881 ;////////////////////////////////////////////////////////////////////////////
            ;3882 ;
            ;3883 T.40.BIT29.NOT.SET.1:
U 1127. 001B,0010,EDB0,0800,0082,130E  ;3884  SC_K[.1B]+1  ; Get ready to mask bit 28
U 130E. 0803,001C,01B0,0800,0000,130F  ;3885  D_NOT.MASK  ; Assert bit 28 in the D register
U 130F. 0001,003C,01B0,09E0,0000,1310  ;3886  RC[0C]_D  ; Save Expected Status of bit 28 in
            ;3887     ; ...Cnfg Reg C/D
            ;3888  Q_D.AND.RC[0A],  ; Mask bit 28 in
U 1310. 0011,0034,01C0,0950,0010,1311  ;3889   CLK.UBCC ; ...CNFG Register C/D
U 1311. 0001,213C,01B0,09D8,0000,1128  ;3890  Z?,RC[0B]_Q  ; Was bit 29 set ?
U 1128. 0000,003C,01B0,0800,0000,112C  ;3891 =0 J/T.40.BIT28.SET.1 ; Yes, continue with the test
U 1129. 0000,003C,0180,0800,0000,112A  ;3892  NOP   ; No, should have been, call an error
```

```
            ;3893  .PAGE
            ;3894 ;
            ;3895 ;////////////////////////////////////////////////////////////////////////////
            ;3896 ;
            ;3897 ; Bit 28 ( Error Log Request ) in configuration register C/D wasn't set and
            ;3898 ; should have been.  Call an error.
            ;3899 ;
            ;3900 =0
U 112A, 0000.003D.D580.0800.0084.709E  ;3901  ERROR2[.6]
U 112B, 0000.003C.0180.0800.0000.112C  ;3902  NOP
            ;3903 ;
            ;3904 ;////////////////////////////////////////////////////////////////////////////
            ;3905 ;
            ;3906 ; ERROR LOGOUT:
            ;3907 ;
            ;3908 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3909 ;  I/O Address of Cnfg Reg C/D of Controller inder test
            ;3910 ;  Expected Status of bit 28 in Cnfg Reg C/D
            ;3911 ;  Received Status of bit 28 in Cnfg Reg C/D
            ;3912 ;  Contents of Configuration Register C/D
            ;3913 ;  Not Used
            ;3914 ;
            ;3915 ;////////////////////////////////////////////////////////////////////////////
            ;3916 ;
            ;3917 =0
            ;3918 T.40.BIT28.SET.1:
U 112C, 0000.003D.0180.0800.0000.1228  ;3919  ERLOOP   ; Loop on error from here
U 112D, 0000.003C.0180.0800.0000.112E  ;3920  NOP
U 112E, 0000.003D.0180.0800.0000.10AC  ;3921 =0 CALL[SBI.INIT]  ; Initialize the SBI
            ;3922  D_R[1],   ; Get data for second error to force
U 112F, 0800.003C.01F8.0A08.0000.1130  ;3923   Q_0  ; Zero the Q register
U 1130, 0000.003D.0180.0800.0000.12D5  ;3924 =0 CALL[FORCE.CRD.RDS.READ]; Force the first error
U 1131, 0010.0038.0180.0968.0200.1312  ;3925  VA_RC[0D]  ; Point VA to configuration register c/d
U 1312, 0000.003C.0180.C800.0000.1313  ;3926  D[LONG]_CACHE.P  ; Read CNFG Register C/D
U 1313, 0001.003C.0180.09D0.0000.1314  ;3927  RC[0A]_D  ; Save CNFG C/D data in RC[0A]
U 1314, 0000.003C.5180.0800.0084.7315  ;3928  SC_K[.1E]  ; Get ready to mask bit 29
U 1315, 0000.003C.0580.0800.0084.B316  ;3929  SC_SC-1   ; ...
U 1316, 0803.001C.0180.0800.0000.1317  ;3930  D_NOT.MASK  ; Assert bit 29 in the D register
U 1317, 0001.003C.0180.09E0.0000.1318  ;3931  RC[0C]_D  ; Set Expected Status of Bit 29
            ;3932     ; ...in Cnfg Reg C/D
            ;3933  Q_D.AND.RC[0C],   ; Mask bit 29 in
U 1318, 0011.0034.01C0.0960.0010.1319  ;3934   CLK.UBCC ; ...CNFG Register C/D
U 1319, 0001.213C.0180.09D8.0000.1132  ;3935  Z?,RC[0B]_Q  ; Was bit 29 set ?
U 1132, 0000.003C.0180.0800.0000.1135  ;3936 =0 J/T.40.BIT29.SET.2 ; Yes, continue with the test
U 1133, 0000.003C.0180.0800.0000.1134  ;3937  NOP   ; No, should have been, call an error
```

```
            ;3938  .PAGE
            ;3939 ;
            ;3940 ;////////////////////////////////////////////////////////////////////////////
            ;3941 ;
            ;3942 ; Bit 29 ( High Error Rate ) in configuration register c/d was not set and
            ;3943 ; should have been.  Call an error.
            ;3944 ;
            ;3945 =0
U 1134, 0000,003D,D580,0800,0084,709E  ;3946  ERROR2[.6]
            ;3947 ;
            ;3948 ;////////////////////////////////////////////////////////////////////////////
            ;3949 ;
            ;3950 ; ERROR LOGOUT:
            ;3951 ;
            ;3952 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3953 ;  I/O Address of Cnfg Reg C/D of Controller under test
            ;3954 ;  Expected Status of bit 29 in Cnfg Reg C/D
            ;3955 ;  Received Status of bit 29 in Cnfg Reg C/D
            ;3956 ;  Contents of Configuration Register C/D
            ;3957 ;  Not Used
            ;3958 ;
            ;3959 ;////////////////////////////////////////////////////////////////////////////
            ;3960 ;
            ;3961 T.40.BIT29.SET.2:
U 1135, 001B,0010,ED80,0800,0082,131A  ;3962  SC_K[.1B]+1  ; Get ready to mask bit 28
U 131A, 0803,001C,0180,0800,0000,131B  ;3963  D_NOT.MASK  ; Assert bit 28 in the D register
U 131B, 0001,003C,0180,09E0,0000,131C  ;3964  RC[0C]_D  ; Set Expected Status of bit 28
            ;3965     ; ...in Cnfg Reg C/D
            ;3966  Q_D.AND.RC[0A],  ; Mask bit 28 in
U 131C, 0011,0034,01C0,0950,0010,131D  ;3967   CLK.UBCC ; ...CNFG Register C/D
U 131D, 0001,213C,0180,09D8,0000,1136  ;3968  Z?,RC[0B]_Q  ; Was bit 28 set ?
U 1136, 0000,003C,0180,0800,0000,113A  ;3969 =0 J/T.40.BIT28.SET.2 ; Yes, continue with the test
U 1137, 0000,003C,0180,0800,0000,1138  ;3970  NOP   ; No, should have been, call an error
```

```
            ;3971   .PAGE
            ;3972 ;
            ;3973 ;////////////////////////////////////////////////////////////////////////////
            ;3974 ;
            ;3975 ; Bit 28 ( Error Log Request ) in configuration register c/d was not set and
            ;3976 ; should have been.  Call an error.
            ;3977 ;
            ;3978 =0
U 1138. 0000,003D,D580,0800,0084,709E   ;3979  ERROR2[.6]
U 1139. 0000,003C,0180,0800,0000,113A   ;3980  NOP
            ;3981 ;
            ;3982 ;////////////////////////////////////////////////////////////////////////////
            ;3983 ;
            ;3984 ; ERROR LOGOUT:
            ;3985 ;
            ;3986 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3987 ;  I/O Address of Cnfg Reg C/D of Controller under test
            ;3988 ;  Expected Status of bit 28 in Cnfg Reg C/D
            ;3989 ;  Received Status of bit 28 in Cnfg Reg C/D
            ;3990 ;  Contents of Configuration Register C/D
            ;3991 ;  Not Used
            ;3992 ;
            ;3993 ;////////////////////////////////////////////////////////////////////////////
            ;3994 ;
            ;3995 =0
            ;3996 T.40.BIT28.SET.2:
U 113A. 0000,003D,0180,0800,0000,1228   ;3997  ERLOOP    ; Loop on error from here
U 113B. 0000,003C,0180,0800,0000,113C   ;3998  NOP
U 113C. 0000,003D,0180,0800,0000,10AC   ;3999 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 113D. 0000,003C,0180,0800,0000,113E   ;4000  NOP
U 113E. 0000,003D,0180,0800,0000,12B5   ;4001 =0 CALL[MS780E.CLEANUP] ; Clear out the Configuration Registers
U 113F. 0010,0038,0180,0968,0200,131E   ;4002  VA_RC[0D]   ; Point VA to Configuration Register C/D
U 131E. 0000,003C,0180,C800,0000,131F   ;4003  D[LONG]_CACHE.P   ; Read Configuration Register C/D
U 131F. 0001,003C,0180,09D0,0000,1320   ;4004  RC[0A]_D   ; Save read data in RC[0A]
U 1320. 0003,003C,0180,09E0,0000,1321   ;4005  RC[0C]_0   ; Set expected Status of bits 28 and 29
            ;4006      ; ...in Cnfg Reg C/D
U 1321. 001B,0010,ED80,0800,0082,1322   ;4007  SC_K[.1B]+1   ; Get ready to mask bit 28
U 1322. 0803,001C,0180,0800,0000,1323   ;4008  D_NOT.MASK   ; Assert bit 28 in the D register and
U 1323. 0000,003C,01E0,0800,0000,1324   ;4009  Q_D    ; Save in Q
U 1324. 0000,003C,0180,0800,0080,D325   ;4010  SC_SC+1    ; Get ready to mask bit 29
U 1325. 0803,001C,0180,0800,0000,1326   ;4011  D_NOT.MASK   ; Assert bit 29 in the D register
U 1326. 081D,0030,0180,0800,0000,1327   ;4012  D_D.OR.Q   ; Now Or D with Q so bit 28 & 29 are set
            ;4013   Q_D.AND.RC[0A],   ; Mask bits 28:29 in
U 1327. 0011,0034,01C0,0950,0010,1328   ;4014    CLK.UBCC  ; ...Configuration register C/D
U 1328. 0001,213C,0180,09D8,0000,1140   ;4015  Z?,RC[0B]_Q   ; Were bits 28 and 29 clear ?
U 1140. 0000,003C,0180,0800,0000,1142   ;4016 =0 J/T.40.28.29.NOT.CLEAR.2; No, should have been, call an error
U 1141. 0000,003E,0180,0800,0000,0001   ;4017  RETURN1    ; Yes, that is the end of call, return to
            ;4018      ; the caller
```

```
            ;4019  .PAGE
            ;4020 ;
            ;4021 ;////////////////////////////////////////////////////////////////////////////////
            ;4022 ;
            ;4023 ; Either of bits 28 and 29 were set in configuration Register C/D.
            ;4024 ; Call an error.
            ;4025 ;
            ;4026 =0
            ;4027 T.40.28.29.NOT.CLEAR.2:
U 1142. 0000.003D.D580.0800.0084.709E  ;4028  ERROR2[.6]
U 1143. 0000.003E.0180.0800.0000.0001  ;4029  RETURN1
            ;4030 ;
            ;4031 ;////////////////////////////////////////////////////////////////////////////////
            ;4032 ;
            ;4033 ; ERROR LOGOUT:
            ;4034 ;
            ;4035 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4036 ;  I/O Address of Cnfg Reg C/D of Controller under test
            ;4037 ;  Expected Status of bits 28 and 29 in Cnfg Reg C/D
            ;4038 ;  Received Status of bits 28 and 29 in Cnfg Reg C/D
            ;4039 ;  Contents of Configuration Register C/D
            ;4040 ;  Not Used
            ;4041 ;
            ;4042 ;////////////////////////////////////////////////////////////////////////////////
            ;4043 ;
            ;4044  ;
            ;4045  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4046  ;
            ;4047 =0
            ;4048 T.40.EXIT:
U 1144. 0000.003D.0180.0800.0000.12B5  ;4049  CALL[MS780E.CLEANUP]  ; Clear Memory CNFG Regs before
U 1145. 0000.003C.0180.0800.0000.1146  ;4050  NOP    ; ...exitting
            ;4051
            ;4052
```

```
;4053  .PAGE "TEST 1C9 ERROR LOGGING (CRD/RDS) TEST"
;4054 ;************************************************************
;4055 ;+++
;4056 ;
;4057 ; Functional Description:
;4058 ; ----------------------
;4059 ;
;4060 ; This test will force similar set of errors as the previous one.
;4061 ; Except this time, the functionality of the syndrome error logging
;4062 ; bits (CNFG Reg C/D bits <6:0>) and CRD, RDS flags (bits <10:9>) in
;4063 ; the same register) will be tested.
;4064 ;
;4065 ;   [Subtest 1:]
;4066 ;
;4067 ; In this subtest it is expected that the error syndromes for the first
;4068 ; CRD error will be logged and that the correct flag is set( bit 9).
;4069 ;
;4070 ;   [Subtest 2:]
;4071 ;
;4072 ; In this subtest it is expected that the first error is logged in the
;4073 ; syndrome bits, and that the correct flag is set (bit 10).
;4074 ;
;4075 ;   [Subtest 3:]
;4076 ;
;4077 ; This subtest expects that the syndrome bits log the RDS error and
;4078 ; ignore the CRD.  Also the RDS error should clear the CRD error flag
;4079 ; and set the RDS flag.
;4080 ;
;4081 ;   [Subtest 4:]
;4082 ;
;4083 ; It is expected in this subtest that the syndrome logged will be those
;4084 ; for the RDS error and that the CRD error does not over-write the error
;4085 ; data for the RDS
;4086 ;
;4087 ; Logic Description:
;4088 ; -----------------
;4089 ; N/A
;4090 ;
;4091 ;
;4092 ; Error Logout:
;4093 ; ------------
;4094 ;
;4095 ; See individual error reports.
;4096 ;
;4097 ; Sync Point Description:
;4098 ; ----------------------
;4099 ; N/A
;4100 ;
;4101 ;============================================================
;4102 ;
;4103 ; Register Map:
;4104 ; ------------
;4105 ;
;4106 ;     RA,RB     Description:
;4107 ;     -----     -----------
```

```
            ;4108 ;
            ;4109 ;       0       First Error Data
            ;4110 ;
            ;4111 ;       1       Second Error Data
            ;4112 ;
            ;4113 ;       2 Expected Syndrome
            ;4114 ;
            ;4115 ;       3 Expecte Error Flag(s) to be Set
            ;4116 ;
            ;4117 ;      RC       Description:
            ;4118 ;      --       -----------
            ;4119 ;
            ;4120 ;       E I/O Base Address of MS780-E Currently Under Test
            ;4121 ;
            ;4122 ;       D I/O Address of Configuration Register C/D
            ;4123 ;
            ;4124 ;       C Contents of Configuratin Register C/D
            ;4125 ;
            ;4126 ; B Received Syndrome
            ;4127 ;
            ;4128 ;--
            ;4129 ;*************************************************************
            ;4130 =0
U 1146, 0000.003D.0180.0800.C000.1093  ;4131 T.41: NEWTST   ; Signify start of newtest
U 1147, 0000.003C.0180.0800.0000.1048  ;4132  NOP
U 1048, 0000.003D.0180.0800.0000.10B2  ;4133 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1049, 0000.003C.0180.0800.0000.1176  ;4134  J/T.41.EXIT  ; No MS780-E's on the SBI to test, exit test
U 104A, 0000.003C.0180.0800.0000.1050  ;4135  NOP
            ;4136 =
            ;4137  ;
            ;4138  ; This is the entry point where the test is restarted in order to
            ;4139  ; test the next controller on the SBI.  If there is no controller
            ;4140  ; out there to test then the test is exited.
            ;4141  ;
            ;4142 =00
            ;4143 RESTART.TEST.41:
U 1050, 0000.003D.0180.0800.0000.10A2  ;4144  CALL[START.TEST] ; Configure the Controllers
U 1051, 0000.003C.0180.0800.0000.1176  ;4145  J/T.41.EXIT  ; No more controllers to configure, exit test
U 1052, 0000.003D.0180.0800.0000.1228  ;4146  ERLOOP   ; Loop on error from here
U 1053, 0000.003C.0180.0800.0000.1148  ;4147  NOP
U 1148, 0000.003D.0180.0800.0000.10AC  ;4148 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1149, 0000.003C.0180.0800.0000.114A  ;4149  NOP
U 114A, 0000.003D.0180.0800.0000.12B5  ;4150 =0 CALL[MS780E.CLEANUP] ; Clear all MS780-E registers
U 114B, 0010.0038.0180.0968.0200.1329  ;4151  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 1329, 0000.003C.0180.C800.0000.132A  ;4152  D[LONG]_CACHE.P  ; Read the contents of CNFG Register C/D
U 132A, 0001.003C.0180.09E0.0000.132B  ;4153  RC[0C]_D  ; Save in RC[0C]
U 132B, 0000.003C.D980.0800.0084.732C  ;4154  SC_K[.9]  ; Get ready to mask bit 9
U 132C, 0803.001C.0180.0800.0000.132D  ;4155  D_NOT.MASK  ; Assert bit 9 in the D register and
U 132D, 0000.003C.01E0.0800.0000.132E  ;4156  Q_D   ; Save in Q
U 132E, 0000.003C.0180.0800.0080.D32F  ;4157  SC_SC+1   ; Get ready to mask bit 10
U 132F, 0803.001C.0180.0800.0000.1330  ;4158  D_NOT.MASK  ; Assert bit 10 in the D register
U 1330, 081D.0030.0180.0800.0000.1331  ;4159  D_D.OR.Q  ; Now Or D with Q so bit 9 & 10 are set
            ;4160  ALU_D[AND]RC[0C], ; Exclusive Or Mask (bits 9:10 set) with
U 1331, 0011.0034.0180.0960.0010.1332  ;4161  CLK.UBCC  ; configuration register C/D data
U 1332, 0000.013C.0180.0800.0000.114C  ;4162  Z?   ; Were bits 9 and 10 clear ?
```

```
U 114C, 0000,003C,0180,0800,0000,114E  ;4163 =0 J/T.41.9.10.NOT.CLEAR.1 ; No, should have been, call an error
U 114D, 0000,003C,0180,0800,0000,114F  ;4164  J/T.41.9.10.CLEAR.1 ; Yes, continue with the test
```

```
            ;4165  .PAGE
            ;4166 ;
            ;4167 ;////////////////////////////////////////////////////////////////////////////////
            ;4168 ;
            ;4169 ; Either/Or bits 9 and 10 were set in configuration Register C/D.
            ;4170 ; Call an error.
            ;4171 ;
            ;4172 =0
            ;4173 T.41.9.10.NOT.CLEAR.1:
U 114E, 0000.003D.0D80.0800.0084.709E  ;4174  ERROR2[.3]
            ;4175 ;
            ;4176 ;////////////////////////////////////////////////////////////////////////////////
            ;4177 ;
            ;4178 ; ERROR LOGOUT:
            ;4179 ;
            ;4180 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4181 ;  I/O Address of Configuration Register C/D
            ;4182 ;  Contents of Configuration Register C/D
            ;4183 ;
            ;4184 ;////////////////////////////////////////////////////////////////////////////////
            ;4185 ;
            ;4186 T.41.9.10.CLEAR.1:
            ;4187 ;
            ;4188 ;================================================================================
            ;4189 ;
            ;4190 ; Subtest 1
            ;4191 ;
U 114F, 0018.0038.0580.0A80.0000.1333  ;4192  R[0]_K[.1]  ; First error will be a CRD
U 1333, 0018.0038.0980.0A88.0000.1334  ;4193  R[1]_K[.2]  ; Second error will be a CRD
U 1334, 0818.0038.8D80.0800.0000.1335  ;4194  D_K[.1F]  ; Load D with x1F
U 1335, 0018.0038.79C0.0800.0000.1336  ;4195  Q_K[.30]  ; Load Q with x30
U 1336, 001D.0014.0180.0A90.0000.1337  ;4196  ALU_D+Q,R[2]_ALU ; Load R[2] with constant x4F, this is the
            ;4197      ; expected syndrome bits
U 1337, 0000.003C.D980.0800.0084.7338  ;4198  SC_K[.9]  ; Get ready to assert Bit 9
U 1338, 0003.001C.0180.0A98.0000.1150  ;4199  R[3]_NOT.MASK  ; Asset Bit 9 in R[3] for constant x200, this
            ;4200      ; is the error flag bit expected to be set
U 1150, 0000.003D.0180.0800.0000.1158  ;4201 =0 CALL[TEST.41.TST] ; Execute the test
            ;4202 ;
            ;4203 ;================================================================================
            ;4204 ;
            ;4205 ; Subtest 2
            ;4206 ;
U 1151, 0018.0038.0D80.0A80.0000.1339  ;4207  R[0]_K[.3]  ; First error will be a RDS
U 1339, 0018.0038.D580.0A88.0000.133A  ;4208  R[1]_K[.6]  ; Second error will also be a RDS
U 133A, 0818.0038.1180.0800.0000.133B  ;4209  D_K[.4]  ; Load D with a constant of x4
U 133B, 0819.0014.0580.0800.0000.133C  ;4210  D_D+K[.1]  ; Add one to get constant of x5
U 133C, 0001.003C.0180.0A90.0000.133D  ;4211  R[2]_D  ; Load R[2] with x5, this is the expected
            ;4212      ; syndromes
U 133D, 0818.0038.8180.0800.0000.133E  ;4213  D_K[.3FF]  ; Load D with a constant of x3FF
U 133E, 0819.0014.0580.0800.0000.133F  ;4214  D_D+K[.1]  ; Add one to get constant of x400
U 133F, 0001.003C.0180.0A98.0000.1152  ;4215  R[3]_D  ; Load R[3] with x400, this is the expected
            ;4216      ; error flag to be set
U 1152, 0000.003D.0180.0800.0000.1158  ;4217 =0 CALL[TEST.41.TST] ; Execute the test
            ;4218 ;
            ;4219 ;================================================================================
```

```
            ;4220 ;
            ;4221 ; Subtest 3
            ;4222 ;
U 1153. 0018,0038,0580,0A80,0000,1340  ;4223  R[0]_K[.1]  ; First error will be a CRD
U 1340. 0018,0038,0D80,0A88,0000,1341  ;4224  R[1]_K[.3]  ; Second error will be a RDS
U 1341. 0818,0038,1180,0800,0000,1342  ;4225  D_K[.4]   ; Load D with a constant of x4
U 1342. 0819,0014,0580,0800,0000,1343  ;4226  D_D+K[.1]  ; Add one to get constant of x5
U 1343. 0001,003C,0180,0A90,0000,1344  ;4227  R[2]_D   ; Load R[2] with x5, this is the expected
            ;4228     ; syndromes
U 1344. 0818,0038,8180,0800,0000,1345  ;4229  D_K[.3FF]  ; Load D with a constant of x3FF
U 1345. 0819,0014,0580,0800,0000,1346  ;4230  D_D+K[.1]  ; Add one to get constant of x400
U 1346. 0001,003C,0180,0A98,0000,1154  ;4231  R[3]_D   ; Load R[3] with x400, this is the expected
            ;4232     ; error flag to be set
U 1154. 0000,003D,0180,0800,0000,1158  ;4233 =0 CALL[TEST.41.TST] ; Execute the test
            ;4234 ;
            ;4235 ;=============================================================================
            ;4236 ;
            ;4237 ; Subtest 4
            ;4238 ;
U 1155. 0018,0038,0D80,0A80,0000,1347  ;4239  R[0]_K[.3]  ; First error will be a RDS
U 1347. 0018,0038,0580,0A88,0000,1348  ;4240  R[1]_K[.1]  ; Second error will be a CRD
U 1348. 0818,0038,1180,0800,0000,1349  ;4241  D_K[.4]   ; Load D with a constant of x4
U 1349. 0819,0014,0580,0800,0000,134A  ;4242  D_D+K[.1]  ; Add one to get constant of x5
U 134A. 0001,003C,0180,0A90,0000,134B  ;4243  R[2]_D   ; Load R[2] with x5, this is the expected
            ;4244     ; syndromes
U 134B. 0818,0038,8180,0800,0000,134C  ;4245  D_K[.3FF]  ; Load D with a constant of x3FF
U 134C. 0819,0014,0580,0800,0000,134D  ;4246  D_D+K[.1]  ; Add one to get constant of x400
U 134D. 0001,003C,0180,0A98,0000,1156  ;4247  R[3]_D   ; Load R[3] with x400, this is the expected
            ;4248     ; error flag to be set
U 1156. 0000,003D,0180,0800,0000,1158  ;4249 =0 CALL[TEST.41.TST] ; Execute the test
U 1157. 0000,003C,0180,0800,0000,1050  ;4250  J/RESTART.TEST.41 ; Restart this test for the next controller
            ;4251  ;
            ;4252  ; To decrease the amount of space used by this test the test itself
            ;4253  ; will be set up as a subroutine.  The parameters that should be passed
            ;4254  ; from the calling routine are the type of errors to be forced, the
            ;4255  ; expected error syndromes and the error bit that should be set in
            ;4256  ; CNFG Register C/D (bit 10 or 9 for RDS and CRD respectively).  The
            ;4257  ; following code is that subroutine
            ;4258  ;
            ;4259 =0
            ;4260 TEST.41.TST:
U 1158. 0000,003D,0180,0800,0000,1230  ;4261  SUBTEST   ; Update the subtest counter
U 1159. 0000,003C,0180,0800,0000,115A  ;4262  NOP
U 115A. 0000,003D,0180,0800,0000,12B5  ;4263 =0 CALL[MS780E.CLEANUP] ; Clear all the error bits
U 115B. 0000,003C,0180,0800,0000,115C  ;4264  NOP
U 115C. 0000,003D,0180,0800,0000,1228  ;4265 =0 ERLOOP   ; loop on error from here
U 115D. 0000,003C,0180,0800,0000,115E  ;4266  NOP
U 115E. 0000,003D,0180,0800,0000,10AC  ;4267 =0 CALL[SBI.INIT]  ; Initialize the SBi
U 115F. 0800,003C,0180,0A00,0000,134E  ;4268  D_R[0]   ; Get data for first error to force
U 134E. 0000,003C,01F8,0800,0000,1160  ;4269  Q_0   ; clear the Q register
U 1160. 0000,003D,0180,0800,0000,12D5  ;4270 =0 CALL[FORCE.CRD.RDS.READ]; Force the first error
U 1161. 0800,003C,0180,0A08,0000,134F  ;4271  D_R[1]   ; Get data to force second error
U 134F. 0000,003C,01F8,0800,0000,1162  ;4272  Q_0   ; clear the Q register
U 1162. 0000,003D,0180,0800,0000,12D5  ;4273 =0 CALL[FORCE.CRD.RDS.READ]; Force the second error
U 1163. 0010,0038,0180,0968,0200,1350  ;4274  VA_RC[0D]  ; Point VA to Configuration Register C/D
```

```
ZZ-ESKAR-V22  TEST 1C9 ERROR LOGGING (CRD/RDS) TEST                                    SEQ 1426
ESKAR4C.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:13                  Page  11

U 1350. 0000.003C.0180.C800.0000.1351  ;4275  D[LONG]_CACHE.P  ; Read Configuration Register C/D
U 1351. 0001.003C.0180.09E0.0000.1352  ;4276  RC[0C]_D  ; Save CNFG REgister C/D data in RC[0C]
U 1352. 0000.003C.01C0.0A18.0000.1353  ;4277  Q_R[3]   ; Load Q with expected error bit to be set
        ;4278  ALU_D.AND.Q,  ; AND expected error bit to be set with
U 1353. 001D.0034.0180.0800.0010.1354  ;4279  CLK.UBCC  ; CNFG register C/D
U 1354. 0000.013C.0180.0800.0000.1164  ;4280  Z?   ; Was the bit set ?
U 1164. 0000.003C.0180.0800.0000.1167  ;4281 =0 J/T.41.ERROR.BIT.SET ; Yes, continue with the test
U 1165. 0000.003C.0180.0800.0000.1166  ;4282  NOP   ; No, should have been, call an error
```

```
ZZ-ESKAR-V22  TEST 1C9 ERROR LOGGING (CRD/RDS) TEST                                                    SEQ 1427
ESKAR4C.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:13                                  Page  11

            ;4283  .PAGE
            ;4284 ;
            ;4285 ;////////////////////////////////////////////////////////////////////////////////
            ;4286 ;
            ;4287 ; Error bit (point to by R[3] ) was not set in Configuration Register C/D.
            ;4288 ; Call an error.
            ;4289 ;
            ;4290 =0
U 1166, 0000,003D,0D80,0800,0084,709E  ;4291  ERROR2[.3]
            ;4292 ;
            ;4293 ;////////////////////////////////////////////////////////////////////////////////
            ;4294 ;
            ;4295 ; ERROR LOGOUT:
            ;4296 ;
            ;4297 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4298 ;  I/O Address of Configuration Register C/D
            ;4299 ;  Contents of Configuration Register C/D
            ;4300 ;
            ;4301 ;////////////////////////////////////////////////////////////////////////////////
            ;4302 ;
            ;4303 T.41.ERROR.BIT.SET:
U 1167, 0810,0038,0180,0960,0000,1355  ;4304  D_RC[0C]  ; Get ready to mask out syndrome bits from
            ;4305      ; configuration register c/d
            ;4306  ALU_D.AND.K[.7F],  ; AND the value of configuration register with
U 1355, 0019,0034,5980,09D8,0000,1356  ;4307  RC[0B]_ALU  ; x7F to mask syndrome bits, save in RC[0B]
U 1356, 0810,0038,0180,0958,0000,1357  ;4308  D_RC[0B]  ; Load D with the received syndrome
            ;4309  ALU_D.XOR.R[2],  ; Exclusive Or received syndrome with the
U 1357, 000D,0020,0180,0A10,0010,1358  ;4310  CLK.UBCC  ; Expected syndrome.
U 1358, 0000,013C,0180,0800,0000,1168  ;4311  Z?    ; Are they equal ?
U 1168, 0000,003C,0180,0800,0000,116A  ;4312 =0 J/T.41.SYNDROME.ERROR ; No, should have been, call an error
U 1169, 0000,003C,0180,0800,0000,116C  ;4313  J/T.41.SYNDROME.EQUAL ; Yes, continue with the test
```

```
            ;4314  .PAGE
            ;4315 ;
            ;4316 ;////////////////////////////////////////////////////////////////////////////
            ;4317 ;
            ;4318 ; Received syndrome and expected syndrome did not match.  Call an error.
            ;4319 ;
            ;4320 =0
            ;4321 T.41.SYNDROME.ERROR:
U 116A, 0000.003D.1180.0800.0084.709E  ;4322  ERROR2[.4]
U 116B, 0000.003C.0180.0800.0000.116C  ;4323  NOP
            ;4324 ;
            ;4325 ;////////////////////////////////////////////////////////////////////////////
            ;4326 ;
            ;4327 ; ERROR LOGOUT:
            ;4328 ;
            ;4329 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4330 ;  I/O Address of Configuration Register C/D
            ;4331 ;  Contents of Configuration Register C/D
            ;4332 ;  Received Syndrome
            ;4333 ;
            ;4334 ;////////////////////////////////////////////////////////////////////////////
            ;4335 ;
            ;4336 =0
            ;4337 T.41.SYNDROME.EQUAL:
U 116C, 0000.003D.0180.0800.0000.1228  ;4338  ERLOOP   ; Loop on error from here
U 116D, 0000.003C.0180.0800.0000.116E  ;4339  NOP
U 116E, 0000.003D.0180.0800.0000.10AC  ;4340 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 116F, 0000.003C.0180.0800.0000.1170  ;4341  NOP
U 1170, 0000.003D.0180.0800.0000.12B5  ;4342 =0 CALL[MS780E.CLEANUP] ; Clear all MS780-E registers
U 1171, 0010.003B.0180.0968.0200.1359  ;4343  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 1359, 0000.003C.0180.C800.0000.135A  ;4344  D[LONG]_CACHE.P  ; Read the contents of CNFG Register C/D
U 135A, 0001.003C.0180.09E0.0000.135B  ;4345  RC[0C]_D  ; Save in RC[0C]
U 135B, 0000.003C.D980.0800.0084.735C  ;4346  SC_K[.9]  ; Get ready to mask bit 9
U 135C, 0803.001C.0180.0800.0000.135D  ;4347  D_NOT.MASK  ; Assert bit 9 in the D register and
U 135D, 0000.003C.01E0.0800.0000.135E  ;4348  Q_D   ; Save in Q
U 135E, 0000.003C.0180.0800.0080.D35F  ;4349  SC_SC+1   ; Get ready to mask bit 10
U 135F, 0803.001C.0180.0800.0000.1360  ;4350  D_NOT.MASK  ; Assert bit 10 in the D register
U 1360, 081D.0030.0180.0800.0000.1361  ;4351  D_D.OR.Q  ; Now Or D with Q so bit 9 & 10 are set
            ;4352  ALU_D[AND]RC[0C], ; "AND" Mask (bits 9:10 set) with
U 1361, 0011.0034.0180.0960.0010.1362  ;4353  CLK.UBCC  ; ...configuration register C/D data
U 1362, 0000.013C.0180.0800.0000.1172  ;4354  Z?   ; Were bits 9 and 10 clear ?
U 1172, 0000.003C.0180.0800.0000.1174  ;4355 =0 J/T.41.9.10.NOT.CLEAR.2 ; No, should have been, call an error
U 1173, 0000.003E.0180.0800.0000.0001  ;4356  RETURN1   ; Yes, done, return to the caller
```

```
            ;4357  .PAGE
            ;4358 ;
            ;4359 ;////////////////////////////////////////////////////////////////////////////
            ;4360 ;
            ;4361 ; Either/Or bits 9 and 10 were set in configuration Register C/D.
            ;4362 ; Call an error.
            ;4363 ;
            ;4364 =0
            ;4365 T.41.9.10.NOT.CLEAR.2:
U 1174, 0000.003D.0D80.0800.0084.709E  ;4366  ERROR2[.3]
            ;4367 ;
            ;4368 ;////////////////////////////////////////////////////////////////////////////
            ;4369 ;
            ;4370 ; ERROR LOGOUT:
            ;4371 ;
            ;4372 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4373 ;  I/O Address of Configuration Register C/D
            ;4374 ;  Contents of Configuration Register C/D
            ;4375 ;
            ;4376 ;////////////////////////////////////////////////////////////////////////////
            ;4377 ;
            ;4378  ;
            ;4379  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4380  ;
U 1175, 0000.003C.0180.0800.0000.1176  ;4381  NOP
            ;4382 =0
            ;4383 T.41.EXIT:
U 1176, 0000.003D.0180.0800.0000.12B5  ;4384  CALL[MS780E.CLEANUP]  ; Clear memory CNFG Regs before
U 1177, 0000.003C.0180.0800.0000.1178  ;4385  NOP     ; ...exitting
            ;4386
            ;4387
```

```
;4388  .PAGE "TEST 1CA ERROR ADDRESS REGISTER TEST"
;4389 ;*************************************************************
;4390 ;***
;4391 ;
;4392 ; Functional Description:
;4393 ; ----------------------
;4394 ;
;4395 ; This test checks that the Error Address bits in CNFG C (or D) register,
;4396 ; bits <27:11>, are loaded with the correct address when an error occurs.
;4397 ; The addresses at which errors are forced are picked in such a manner
;4398 ; that upon reading the error address field, a specific pattern will be
;4399 ; returned.
;4400 ;
;4401 ;
;4402 ; Logic Description:
;4403 ; -----------------
;4404 ; N/A
;4405 ;
;4406 ; Error Logout:
;4407 ; ------------
;4408 ;
;4409 ; See individual error reports.
;4410 ;
;4411 ; Sync Point Description:
;4412 ; ----------------------
;4413 ; N/A
;4414 ;
;4415 ;=============================================================
;4416 ;
;4417 ; Register Map:
;4418 ; ------------
;4419 ;
;4420 ;     RA,RB     Description:
;4421 ;     -----     -----------
;4422 ;
;4423 ;       0       Saved Memory Size
;4424 ;
;4425 ;       1       64K/256K Flag (1 = 64K, 0 = 256K)
;4426 ;
;4427 ;       2 Board Counter
;4428 ;
;4429 ;       3 Bank Counter
;4430 ;
;4431 ;      RC       Description:
;4432 ;      --       -----------
;4433 ;
;4434 ;       E I/O Base Address of MS780-E Currently Under Test
;4435 ;
;4436 ;       D I/O Address of Configuration Regsiter C/D
;4437 ;
;4438 ;       C Expected Value of Bits Under Test
;4439 ;
;4440 ;       B Received Value of Bits Under Test
;4441 ;
;4442 ;       A Contents of Configuration Register C/D
```

```
            ;4443 ;
            ;4444 ;       9 Bank Counter
            ;4445 ;
            ;4446 ; 8 Board Pointer
            ;4447 ;
            ;4448 ;--
            ;4449 ;*************************************************************
            ;4450 =0
U 1178, 0000,003D,0180,0800,0000,1093  ;4451 T.42: NEWTST   ; Signify start of new test
U 1179, 0000,003C,0180,0800,0000,1054  ;4452  NOP
U 1054, 0000,003D,0180,0800,0000,10B2  ;4453 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the system
U 1055, 0000,003C,0180,0800,0000,119C  ;4454  J/T.42.EXIT  ; No MS780-E on this system, exit test.
U 1056, 0000,003C,0180,0800,0000,1060  ;4455  NOP
            ;4456 =
            ;4457  ;
            ;4458  ; This entry point is the place where the test is completely restarted
            ;4459  ; to test the next controller.  If there is no more controllers then
            ;4460  ; the test is exited.
            ;4461  ;
            ;4462 =00
            ;4463 RESTART.TEST.42:
U 1060, 0000,003D,0180,0800,0000,10A2  ;4464  CALL[START.TEST] ; Configure the controllers
U 1061, 0000,003C,0180,0800,0000,119C  ;4465  J/T.42.EXIT  ; No more MS780-E's on the system, exit
U 1062, 0000,003C,0180,0800,0000,1064  ;4466  NOP
            ;4467 =
U 1064, 0000,003D,0180,0800,0000,12C9  ;4468 =00 CALL[64K.OR.256K] ; Find out what kind of memory arrays are
            ;4469      ; present (64K or 256K)
U 1065, 0000,003C,0180,0800,0000,117A  ;4470  J/T.42.MIX.ERROR ; CASE 1:  Return here when there is a mixture
            ;4471      ; of arrays on the controller or there were
            ;4472      ; no arrays found.
U 1066, 0000,003C,0180,0800,0000,117B  ;4473  J/T.42.64K.CHIPS ; CASE 2:  Return here when there is 64k ram
            ;4474      ; array boards present (1 mega byte array)
U 1067, 0000,003C,0180,0800,0000,1363  ;4475  J/T.42.256K.CHIPS ; CASE 3:  Return here when there is 256k ram
            ;4476      ; array boards present (4 mega byte array)
```

```
            ;4477  .PAGE
            ;4478 ;
            ;4479 ;////////////////////////////////////////////////////////////////////////////
            ;4480 ;
            ;4481 ; Got an error from 64K.OR.256K subroutine.  Either there are no arrays present
            ;4482 ; or there are a mixture of 64K and 256K arrays.  Call an error.
            ;4483 ;
            ;4484 =0
            ;4485 T.42.MIX.ERROR:
U 117A, 0000,003D,0980,0800,0084,709C  ;4486  ERROR1[.2]
            ;4487 ;
            ;4488 ;////////////////////////////////////////////////////////////////////////////
            ;4489 ;
            ;4490 ; ERROR LOGOUT:
            ;4491 ;
            ;4492 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4493 ;  Configuration Register C or D ( C = Lower & D = Upper)
            ;4494 ;
            ;4495 ;////////////////////////////////////////////////////////////////////////////
            ;4496 ;
            ;4497  ;
            ;4498  ; Come here from routine 64K.OR.256K when the memory chips are 64K
            ;4499  ;
            ;4500 T.42.64K.CHIPS:
            ;4501  R[01]_K[.1],  ; Set flag to indicate that 64k chips present
U 117B, 0018,0038,0580,0A88,0000,117C  ;4502  J/T.42.GET.MEMORY.SIZE ; Got get memory size
            ;4503  ;
            ;4504  ; Come here from routine 64K.OR.256K when the memory chips are 256K
            ;4505  ;
            ;4506 T.42.256K.CHIPS:
U 1363, 0003,003C,0180,0A88,0000,117C  ;4507  R[01]_0   ; Clear flag to indicate that 256k chips
            ;4508      ; are present
            ;4509 =0
            ;4510 T.42.GET.MEMORY.SIZE:
U 117C, 0000,003D,0180,0800,0000,12CE  ;4511  CALL[GET.MEMORY.SIZE] ; Get the memory size (returned in the D reg)
U 117D, 0001,003C,0180,0A80,0000,1364  ;4512  R[0]_D   ; Save memory size in R[0]
U 1364, 0003,003C,0180,09C0,0000,1365  ;4513  RC[08]_0  ; Clear Board Pointer
U 1365, 0003,003C,0180,0A90,0000,1366  ;4514  R[02]_0   ; Clear Board Counter ;
            ;4515  ; This is the loop point for checking each board.
            ;4516  ;
            ;4517 T.42.LOOP.POINT.ONE:
U 1366, 0F00,003C,0180,0800,0000,1367  ;4518  D_0   ; Clear the D Reg
            ;4519  R[03]_ALU,  ; Clear Bank Counter
U 1367, 0001,0028,0180,0A98,0000,1368  ;4520   ALU_NOT.D ; ...
U 1368, 0810,0038,0180,0940,0000,1369  ;4521  D_RC[08]  ; Move Board pointer to the D register
U 1369, 0003,003C,0180,09C8,0000,117E  ;4522  RC[09]_0  ; Initialize the Bank counter
            ;4523  ;
            ;4524  ; This is the loop point for checking each bank within the board.
            ;4525  ;
            ;4526 =0
            ;4527 T.42.LOOP.POINT.TWO:
U 117E, 0000,003D,0180,0800,0000,1228  ;4528  ERLOOP   ; Loop on error from here
U 117F, 0000,003C,0180,0800,0000,1180  ;4529  NOP
U 1180, 0000,003D,0180,0800,0000,12B5  ;4530 =0 CALL[MS780E.CLEANUP] ; Clear MS780-E/H CNFG Regs
U 1181, 0800,003C,0180,0A18,0000,136A  ;4531  D_R[3]   ; Get Bank Counter
```

```
U 136A. 0019.0014.0580.0A98.0000.1182  ;4532  R[3]_D+K[.1]  ; Increment Bank Counter
U 1182. 0000.003D.0180.0800.0000.10AC  ;4533 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1183. 0000.003C.0180.0800.0000.1184  ;4534  NOP
U 1184. 0818.0039.0980.0800.0000.1243  ;4535 =0 SET.DIAG.MODE[.2] ; Select ECC substitute diagnostic mode
U 1185. 0000.003C.0180.0800.0000.1186  ;4536  NOP
U 1186. 0818.0039.8580.0800.0000.12AF  ;4537 =0 SET.ECC[.C]  ; set substitute check bits for dat of all 0's
U 1187. 0810.0038.0180.0948.0000.136B  ;4538  D_RC[09]  ; Move counter to the D register
U 136B. 0010.0038.01C0.0940.0000.136C  ;4539  Q_RC[08]  ; Move Board pointer to the Q register
        ;4540  VA_Q+D,   ; Point VA to address to force the error
U 136C. 001D.2014.0180.0AB8.0200.136D  ;4541   R[7]_ALU ; Save SUM in R[7]
U 136D. 0818.0038.0D80.0800.0000.136E  ;4542  D_K[.3]   ; Get data to initialize the memory
U 136E. 0000.003C.0180.A800.0000.136F  ;4543  CACHE.P_D[LONG]  ; Initialize the location so that
        ;4544     ; ...a RDS error will be forced
U 136F. 0000.003C.0180.0803.0000.1370  ;4545  VA_VA+4   ; Point to next location
U 1370. 0F00.003C.0180.0800.0000.1371  ;4546  D_0   ; Get data to initialize the next location
U 1371. 0000.003C.0180.A800.0000.1188  ;4547  CACHE.P_D[LONG]  ; Initialize
U 1188. 0000.003D.8580.0800.0084.726B  ;4548 =0 SET.TRAP.MASK[.C] ; Enable RDS uTraps
U 1189. 0000.003C.0180.0A38.0200.1372  ;4549  VA_ALU,ALU_R[7]  ; Point VA again to location
        ;4550     ; ...under test
U 1372. 0000.003C.0180.C800.0000.1373  ;4551  D[LONG]_CACHE.P  ; Force the error
U 1373. 0010.0038.0180.0968.0200.1374  ;4552  VA_RC[0D]  ; Point VA to configuration register c/d
U 1374. 0000.003C.0180.C800.0000.1375  ;4553  D[LONG]_CACHE.P  ; Read the contents of cnfg c/d
U 1375. 0001.003C.0180.09D0.0000.1376  ;4554  RC[0A]_D  ; Save cnfg c/d data in RC[0A]
U 1376. 0818.0038.2180.0800.0000.1377  ;4555  D_K[.14]  ; Form x15 in the D reg
U 1377. 0819.0014.0580.0800.0000.1378  ;4556  D_D+K[.1]  ; ...
        ;4557  ALU_NOT.D,  ; Form NEGATIVE 15 in the SC
U 1378. 0001.0028.0180.0800.0082.1379  ;4558   SC_ALU  ; Get read to shift cnfg c/d data right 14
        ;4559  D_RC[0A],  ; Load D with saved cnfg c/d data
U 1379. 0810.0038.01F8.0950.0000.137A  ;4560   Q_0  ; Zero the Q register as it is the shift in
U 137A. 0D00.003C.0180.0800.0000.137B  ;4561  D_DAL.SC  ; Shift bits 20:21 to 0:1
U 137B. 0019.0034.0D80.09D8.0000.137C  ;4562  RC[0B]_D.AND.K[.3] ; Mask out bits 0:1 to RC[0B]
U 137C. 0800.003C.0180.0A18.0000.137D  ;4563  D_R[3]   ; Load D with expected value bank select bits
U 137D. 0001.003C.0180.09E0.0000.137E  ;4564  RC[0C]_D  ; and save in RC[0C]
        ;4565  ALU_D.XOR.RC[0B], ; Exclusive Or the expected and received bank
U 137E. 0011.0020.0180.0958.0010.137F  ;4566  CLK.UBCC  ; select bits
U 137F. 0000.013C.0180.0800.0000.118A  ;4567  Z?   ; Are they equal ?
U 118A. 0000.003C.0180.0800.0000.118C  ;4568 =0 J/T.42.BANK.SEL.ERROR ; No, Call an error
U 118B. 0000.003C.0180.0800.0000.118D  ;4569  J/T.42.CHK.BOARD.SEL ; Yes, continue with test
```

```
            ;4570  .PAGE
            ;4571 ;
            ;4572 ;////////////////////////////////////////////////////////////////////////////
            ;4573 ;
            ;4574 ; Expected and Received bank select bits did not match.  Call an error.
            ;4575 ;
            ;4576 =0
            ;4577 T.42.BANK.SEL.ERROR:
U 118C, 0000,003D,5D80,0800,0084,709E  ;4578  ERROR2[.7]
            ;4579 ;
            ;4580 ;////////////////////////////////////////////////////////////////////////////
            ;4581 ;
            ;4582 ; ERROR LOGOUT:
            ;4583 ;
            ;4584 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4585 ;  I/O Address of Configuration Register C/D
            ;4586 ;  Expected Value of Bits Under Test
            ;4587 ;  Received Value of Bits Under Test
            ;4588 ;  Contents of Configuration Register C/D
            ;4589 ;  Bank Counter
            ;4590 ;  Board Pointer
            ;4591 ;
            ;4592 ;////////////////////////////////////////////////////////////////////////////
            ;4593 ;
            ;4594 T.42.CHK.BOARD.SEL:
U 118D, 081B,001C,7D80,0800,0000,1380  ;4595  D_NOT.K[.18]  ; Move negative 18 to the D register
U 1380, 0001,003C,0180,0800,0082,1381  ;4596  SC_D   ; Get read to shift cnfg c/d data right x18
            ;4597  D_RC[0A],  ; Load D with saved cnfg c/d data
            ;4598   SC_SC+1, ; Adjust SC for the correct number of shifts
U 1381, 0810,0038,01F8,0950,0080,D382  ;4599   Q_0  ; Zero the Q register as it is the shift in
U 1382, 0D00,003C,0180,0800,0000,1383  ;4600  D_DAL.SC  ; Shift bits 24:26 to 0:2
U 1383, 0019,0034,5D80,09D8,0000,1384  ;4601  RC[0B]_D.AND.K[.7] ; Mask out bits 0:2 to RC[0B]
U 1384, 0800,003C,0180,0A10,0000,1385  ;4602  D_R[2]   ; Load D with expected value board select bits
U 1385, 0001,003C,0180,09E0,0000,1386  ;4603  RC[0C]_D  ; and save in RC[0C]
            ;4604  ALU_D.XOR.RC[0B], ; Exclusive Or the expected and received board
U 1386, 0011,0020,0180,0958,0010,1387  ;4605  CLK.UBCC  ; select bits
U 1387, 0000,013C,0180,0800,0000,118E  ;4606  Z?   ; Are they equal ?
U 118E, 0000,003C,0180,0800,0000,1190  ;4607 =0 J/T.42.BOARD.SEL.ERROR ; No, Call an error
U 118F, 0000,003C,0180,0800,0000,1191  ;4608  J/T.42.CHK.CONTROL.SEL ; Yes, continue with test
```

```
            ;4609  .PAGE
            ;4610 ;
            ;4611 ;////////////////////////////////////////////////////////////////////////////////
            ;4612 ;
            ;4613 ; Expected and Received board select bits did not match.  Call an error.
            ;4614 ;
            ;4615 =0
            ;4616 T.42.BOARD.SEL.ERROR:
U 1190, 0000,003D,5D80,0800,0084,709E  ;4617  ERROR2[.7]
            ;4618 ;
            ;4619 ;////////////////////////////////////////////////////////////////////////////////
            ;4620 ;
            ;4621 ; ERROR LOGOUT:
            ;4622 ;
            ;4623 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4624 ;  I/O Address of Configuration Register C/D
            ;4625 ;  Expected Value of Bits Under Test
            ;4626 ;  Received Value of Bits Under Test
            ;4627 ;  Contents of Configuration Register C/D
            ;4628 ;  Bank Counter
            ;4629 ;  Board Pointer
            ;4630 ;
            ;4631 ;////////////////////////////////////////////////////////////////////////////////
            ;4632 ;
            ;4633 T.42.CHK.CONTROL.SEL:
U 1191, 081B,001C,ED80,0800,0000,1388  ;4634  D_NOT.K[.1B]  ; Move negative 1B to the D register
U 1388, 0001,003C,0180,0800,0082,1389  ;4635  SC_D   ; Get ready to shift cnfg c/d data right x1b
            ;4636  D_RC[0A],  ; Load D with saved cnfg c/d data
            ;4637   SC_SC+1, ; Adjust SC for correct number of shifts
U 1389, 0810,0038,01F8,0950,0080,D38A  ;4638   Q_0  ; Zero the Q register as it is the shift in
U 138A, 0D00,003C,0180,0800,0000,138B  ;4639  D_DAL.SC  ; Shift bit 27 to 0
U 138B, 0019,0034,0580,09D8,0000,138C  ;4640  RC[0B]_D.AND.K[.1] ; Mask out bit 0 to RC[0B]
            ;4641  D_RC[0D],  ; Load D with cnfg c/d address
U 138C, 0810,0038,01F8,0968,0000,138D  ;4642   Q_0  ; Zero the Q register as it is the shift in
            ;4643  D_D.RIGHT2,  ; Shift bit 2 to in bit position 0
U 138D, 0200,003C,0180,0800,0000,138E  ;4644   SI/ZERO  ; ...shift in zeroes
U 138E, 0019,0034,0580,09E0,0000,138F  ;4645  RC[0C]_D.AND.K[.1] ; Mask out bit 0 to RC[0C]
U 138F, 0810,0038,0180,0960,0000,1390  ;4646  D_RC[0C]  ; Load D with expected value of this bit
            ;4647  ALU_D.XOR.RC[0B], ; Exclusive Or the expected and received value
U 1390, 0011,0020,0180,0958,0010,1391  ;4648  CLK.UBCC  ; of this bit
U 1391, 0000,013C,0180,0800,0000,1192  ;4649  Z?   ; Are they equal ?
U 1192, 0000,003C,0180,0800,0000,1194  ;4650 =0 J/T.42.CONTROL.SEL.ERROR; No, Call an error
U 1193, 0000,003C,0180,0800,0000,1195  ;4651  J/T.42.UPDATE.CHK.R3 ; Yes, continue with test
```

```
            ;4652  .PAGE
            ;4653 ;
            ;4654 ;////////////////////////////////////////////////////////////////////////////
            ;4655 ;
            ;4656 ; Expected and Received controller select bits did not match.  Call an error.
            ;4657 ;
            ;4658 =0
            ;4659 T.42.CONTROL.SEL.ERROR:
U 1194, 0000,003D,5D80,0800,0084,709E  ;4660  ERROR2[.7]
            ;4661 ;
            ;4662 ;////////////////////////////////////////////////////////////////////////////
            ;4663 ;
            ;4664 ; ERROR LOGOUT:
            ;4665 ;
            ;4666 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4667 ;  I/O Address of Configuration Register C/D
            ;4668 ;  Expected Value of Bits Under Test
            ;4669 ;  Received Value of Bits Under Test
            ;4670 ;  Contents of Configuration Register C/D
            ;4671 ;  Bank Counter
            ;4672 ;  Board Pointer
            ;4673 ;
            ;4674 ;////////////////////////////////////////////////////////////////////////////
            ;4675 ;
            ;4676 T.42.UPDATE.CHK.R3:
U 1195, 0810,0038,0180,0948,0000,1392  ;4677  D_RC[09]  ; Move Bank pointer to the D register
U 1392, 0019,0014,1180,09C8,0000,1393  ;4678  RC[09]_D+K[.4]  ; Update bank pointer and restore in RC[09]
            ;4679  ALU_R[3].XOR.K[.3], ; Exclusive Or the bank pointer with 3 to
U 1393, 0018,0020,0D80,0A18,0010,1394  ;4680  CLK.UBCC  ; determine if done with this bank
U 1394, 0000,013C,0180,0800,0000,1196  ;4681  Z?  ; Are they equal ?
U 1196, 0000,003C,0180,0800,0000,117E  ;4682 =0 J/T.42.LOOP.POINT.TWO ; No, do another loop
U 1197, 0800,003C,0180,0A10,0000,1395  ;4683  D_R[02]  ; Yes, point to another board
U 1395, 0019,0014,0580,0A90,0000,1396  ;4684  R[02]_D+K[.1]  ; Increment the board counter
U 1396, 0000,003C,0180,0A08,0010,1397  ;4685  ALU_R[01],CLK.UBCC ; Move 64K/256K flag to the ALU
U 1397, 0000,013C,0180,0800,0000,1198  ;4686  Z?  ; Are there 64K or 256K chips on array ?
U 1198, 0000,003C,0180,0800,0000,139A  ;4687 =0 J/T.42.DOING.64K ; 64k on array
U 1199, 0000,003C,7D80,0800,0084,7398  ;4688  SC_K[.18]  ; Load x18 into the SC
U 1398, 0000,003C,0980,0800,0084,B399  ;4689  SC_SC-K[.2]  ; subtract 2 so SC has X16 for masking out the
            ;4690     ; value x400000
            ;4691  D_NOT.MASK,  ; Load  the D register with x400000
U 1399, 0803,001C,0180,0800,0000,139C  ;4692  J/T.42.ADD.D.TO.RC8 ; Go add the D register do RC[08]
            ;4693 T.42.DOING.64K:
U 139A, 0000,003C,2180,0800,0084,739B  ;4694  SC_K[.14]  ; Load SC with x14 for masking out the value
            ;4695     ; of x100000
U 139B, 0803,001C,0180,0800,0000,139C  ;4696  D_NOT.MASK  ; Load the D register with x100000
            ;4697 T.42.ADD.D.TO.RC8:
U 139C, 0010,0038,01C0,0940,0000,139D  ;4698  Q_RC[08]  ; Load the Q register with the Board Pointer
U 139D, 001D,0014,0180,09C0,0000,139E  ;4699  RC[08]_D+Q  ; Add the offset to the board pointer
U 139E, 0810,0038,0180,0940,0000,139F  ;4700  D_RC[08]  ; Move board pointer to the D register
            ;4701  ALU_D.XOR.R[00], ; Exclusive Or the board pointer with the
U 139F, 000D,0020,0180,0A00,0010,13A0  ;4702  CLK.UBCC  ; Maximum address
U 13A0, 0000,013C,0180,0800,0000,119A  ;4703  Z?  ; Are they equal ?
U 119A, 0000,003C,0180,0800,0000,1366  ;4704 =0 J/T.42.LOOP.POINT.ONE ; No, restart this loop
U 119B, 0000,003C,0180,0800,0000,1060  ;4705  J/RESTART.TEST.42
            ;4706  ;
```

```
            ;4707  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4708  ;
            ;4709 =0
            ;4710 T.42.EXIT:
U 119C, 0000,003D,0180,0800,0000,12B5  ;4711  CALL[MS780E.CLEANUP] ; Clear Memory CNFG Regs before exitting
U 119D, 0000,003C,0180,0800,0000,119E  ;4712  NOP
            ;4713
            ;4714
```

```
;4715  .PAGE "TEST 1CB SINGLE/DOUBLE BIT ERROR DETECT. ON EXT READS"
;4716 ;*************************************************************
;4717 ;+++
;4718 ;
;4719 ; Functional Description:
;4720 ; ----------------------
;4721 ;
;4722 ; The following sequence of subtests forces errors in both longwords of
;4723 ; a quadword at locations 0 and 4 on each controller, while performing
;4724 ; Extended Reads.  CRD, RDS, HIGH ERROR RATE and ERROR LOG bits in CNFG
;4725 ; C/D are checked to see whether they are set or not (depending on what
;4726 ; the test is supposed to do).
;4727 ;
;4728 ; In all the subtests below, the contents of the SBI.ERR register will
;4729 ; be checked for the appropriate condition (ie., CRD bit set, RDS bit set
;4730 ; or both set).  This will verify that the respective tags were sent on
;4731 ; the SBI, and that the SBI detected them.
;4732 ;
;4733 ;   [Subtest 1:]
;4734 ;
;4735 ; This subtest forces a CRD error in the upper longword on an Extended
;4736 ; Read and verifies normal operation:  CRD tag sent on the SBI, corrected
;4737 ; data returned from memory and that the data corrected on the array.
;4738 ; Also, a check is made to see that in CNFG C/D register, the CRD bit and
;4739 ; the Error logout bit are set while the RDS and High Error Rate bits
;4740 ; are clear.
;4741 ;
;4742 ;   [Subtest 2:]
;4743 ;
;4744 ; A RDS error is forced in the upper longword, only, during an Extended
;4745 ; Read.  It is expected that:  no micro trap will occur, RDS tag sent to
;4746 ; the CPU, data is not corrected on the array and the proper syndrome
;4747 ; bits were generated.  Also CNFG C/D register will be checked to verify
;4748 ; that the RDS and Error Log bits are set while the CRD and High Error
;4749 ; rate are clear.
;4750 ;
;4751 ;   [Subtest 3:]
;4752 ;
;4753 ; This subtest forces a CRD error in the lower longword and a RDS error
;4754 ; in the upper longword.  It is expected that no RDS micro trap will
;4755 ; occur.  Contents of CNFG C/D register are expected to show both RDS
;4756 ; and CRD bits set, as well as the High Error Rate bit and the Error
;4757 ; Log bit set.
;4758 ;
;4759 ;   [Subtest 4:]
;4760 ;
;4761 ; A RDS error is forced in the lower longword and a CRD in the upper
;4762 ; longword on Extended Reads.  A micro-trap should occur for the RDS
;4763 ; error.  The following bits in CNFG C/D register should be set:  RDS,
;4764 ; CRD, High Error Rate, and Error log.
;4765 ;
;4766 ;
;4767 ; Logic Description:
;4768 ; -----------------
;4769 ;
```

```
                ;4770 ;  N/A
                ;4771 ;
                ;4772 ; Error Logout:
                ;4773 ; ------------
                ;4774 ;
                ;4775 ; See individual error reports.
                ;4776 ;
                ;4777 ; Sync Point Description:
                ;4778 ; ----------------------
                ;4779 ;  N/A
                ;4780 ;
                ;4781 ;=============================================================
                ;4782 ;
                ;4783 ; Register Map:
                ;4784 ; ------------
                ;4785 ;
                ;4786 ;     RA,RB     Description:
                ;4787 ;     -----     -----------
                ;4788 ;
                ;4789 ;       0       Bit Mask for Configuration Register C/D
                ;4790 ;
                ;4791 ;       1 Bits Expected to be Set in Configuration Register C/D
                ;4792 ;
                ;4793 ;      RC       Description:
                ;4794 ;      --       -----------
                ;4795 ;
                ;4796 ;       E I/O Base Address of MS780-E Currently Under Test
                ;4797 ;
                ;4798 ;       D I/O Address of Configuration Register C/D
                ;4799 ;
                ;4800 ;       B Configuration Register C/D Bits that are Under Test
                ;4801 ;
                ;4802 ;       A Contents of Configuration Register C/D
                ;4803 ;
                ;4804 ;--
                ;4805 ;*************************************************************
                ;4806 =0
U 119E, 0000,003D,0180,0800,0000,1093  ;4807 T.43: NEWTST   ; Signify the start of a new test
U 119F, 0000,003C,0180,0800,0000,1068  ;4808  NOP
U 1068, 0000,003D,0180,0800,0000,10B2  ;4809 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1069, 0000,003C,0180,0800,0000,120E  ;4810  J/T.43.EXIT  ; No MS780-E's on the SBI to test, exit
U 106A, 0000,003C,0180,0800,0000,106C  ;4811  NOP
                ;4812 =
                ;4813  ;
                ;4814  ; This is the restart point for the test when the current controller
                ;4815  ; is tested completely and it is time to go on to the next controller
                ;4816  ; and if there is no next controller then exit this test.
                ;4817  ;
                ;4818 =00
                ;4819 RESTART.TEST.43:
U 106C, 0000,003D,0180,0800,0000,10A2  ;4820  CALL[START.TEST] ; configure the controllers
U 106D, 0000,003C,0180,0800,0000,120E  ;4821  J/T.43.EXIT  ; no controllers left to configure
U 106E, 0000,003C,5180,0800,0084,706F  ;4822  SC_K[.1E]  ; Load SC with x1D to mask bit 29
U 106F, 0000,003C,0580,0800,0084,B3A1  ;4823   SC_SC-1  ; ...
U 13A1, 0803,001C,0180,0800,0000,13A2  ;4824  D_NOT.MASK  ; Assert bit 29 in the D register
```

```
U 13A2. 0000.003C.0580.0800.0084.B3A3  ;4825  SC_SC-1   ; Update SC to contain x1C to mask bit 28
U 13A3. 0801.001C.0180.0800.0000.13A4  ;4826  D_D.ORNOT.MASK  ; Assert bits 29:28 in the D register
U 13A4. 0000.003C.F580.0800.0084.73A5  ;4827  SC_K[.A]  ; Get ready to mask bit 10
U 13A5. 0801.001C.0180.0800.0000.13A6  ;4828  D_D.ORNOT.MASK  ; Assert bits 29:28 & bit 10 in the D register
U 13A6. 0000.003C.0580.0800.0084.B3A7  ;4829  SC_SC-1   ; Update SC to contain x9 to mask bit 9
U 13A7. 0801.001C.0180.0800.0000.13A8  ;4830  D_D.ORNOT.MASK  ; Assert bits 29:28 & 10:9 in the D register
U 13A8. 0001.003C.0180.0A80.0000.11A0  ;4831  R[0]_D   ; Load R[0] with x30000600 which is the bit
          ;4832     ; mask for Configuration Register C/D
          ;4833 ;
          ;4834 ;================================================================================
          ;4835 ;
          ;4836 ; Subtest 1
          ;4837 ;
U 11A0. 0000.003D.0180.0800.0000.1230  ;4838 =0 SUBTEST   ; Update the Subtest counter
U 11A1. 001B.0010.ED80.0800.0082.13A9  ;4839  SC_K[.1B]+1  ; Load SC with x1C to mask bit 28
U 13A9. 0803.001C.0180.0800.0000.13AA  ;4840  D_NOT.MASK  ; Assert bit 28 in the D register
U 13AA. 0000.003C.D980.0800.0084.73AB  ;4841  SC_K[.9]  ; Get ready to mask bit 9
U 13AB. 0801.001C.0180.0800.0000.13AC  ;4842  D_D.ORNOT.MASK  ; Assert bits 28 & bit 9 in the D register
U 13AC. 0001.003C.0180.0A88.0000.11A2  ;4843  R[01]_D   ; Load R[01] with x10000200 which is the
          ;4844     ; expected set bits in CNFG C/D
U 11A2. 0000.003D.0180.0800.0000.1228  ;4845 =0 ERLOOP   ; loop on error from here
U 11A3. 0000.003C.0180.0800.0000.11A4  ;4846  NOP
U 11A4. 0000.003D.0180.0800.0000.10AC  ;4847 =0 CALL[SBI.INIT]  ; initialize the SBI
U 11A5. 0000.003C.0180.0800.0000.11A6  ;4848  NOP
U 11A6. 0000.003D.0180.0800.0000.12B5  ;4849 =0 CALL[MS780E.CLEANUP] ; Clear memory configuration registers
U 11A7. 0018.0038.05C0.0800.0000.13AD  ;4850  Q_K[.1]   ; Data for error to be forced in upper longword
U 13AD. 0F00.003C.0180.0800.0000.11A8  ;4851  D_0   ; No error forced in lower longword
U 11A8. 0000.003D.0180.0800.0000.12D5  ;4852 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11A9. 0000.003C.0180.0800.0000.1070  ;4853  NOP
U 1070. 0000.003D.0180.0800.0000.1265  ;4854 =00 CHECK.UTRAP   ; Was there a RDS micro trap ?
U 1071. 0000.003C.0180.0800.0000.1073  ;4855  J/T.43.NO.RDS.UTRAP.1 ; No, continue with the test
```

```
            ;4856  .PAGE
            ;4857 ;
            ;4858 ;////////////////////////////////////////////////////////////////////////////
            ;4859 ;
            ;4860 ; Got a RDS micro trap when forcing a CRD in upper longword and shouldn't
            ;4861 ; have.  Call an error.
            ;4862 ;
U 1072, 0000,003D,0180,0800,0084,709E  ;4863  ERROR2[.8]
            ;4864 ;
            ;4865 ;////////////////////////////////////////////////////////////////////////////
            ;4866 ;
            ;4867 ; ERROR LOGOUT:
            ;4868 ;
            ;4869 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4870 ;  I/O Address of Configuration Register C/D
            ;4871 ;  Unused
            ;4872 ;  Configuration Register C/D Bits that are Under Test
            ;4873 ;  Configuration Register C/D Data
            ;4874 ;  Unused
            ;4875 ;  Expected Micro Trap Flags
            ;4876 ;  Received Micro Trap Flags
            ;4877 ;
            ;4878 ;////////////////////////////////////////////////////////////////////////////
            ;4879 ;
            ;4880 T.43.NO.RDS.UTRAP.1:
U 1073, 0010,0038,0180,0968,0200,13AE  ;4881  VA_RC[0D]  ; Point VA to configuration register c/d
U 13AE, 0000,003C,0180,C800,0000,13AF  ;4882  D[LONG]_CACHE.P  ; Read configuration register c/d
U 13AF, 0001,003C,0180,09D0,0000,13B0  ;4883  RC[0A]_D  ; Load RC[0A] with contents of cnfg c/d
U 13B0, 081C,2034,0180,0A00,0000,13B1  ;4884  D_D.AND.R[0]  ; Mask out the unwanted bits and then
U 13B1, 0001,003C,0180,09D8,0000,13B2  ;4885  RC[0B]_D  ; store in RC[0B]
            ;4886  ALU_D.XOR.R[1],  ; Exclusive Or expected error bits with the
U 13B2, 000D,0020,0180,0A08,0010,13B3  ;4887  CLK.UBCC  ; received error bits
U 13B3, 0000,013C,0180,0800,0000,11AA  ;4888  Z?  ; Were the correct error bits set ?
U 11AA, 0000,003C,0180,0800,0000,11AC  ;4889 =0 J/T.43.WRONG.BITS.SET.1 ; No, call an error
U 11AB, 0000,003C,0180,0800,0000,11AE  ;4890  J/T.43.BITS.OK.1 ; yes, continue with the test
```

```
            ;4891   .PAGE
            ;4892 ;
            ;4893 ;////////////////////////////////////////////////////////////////////////////////
            ;4894 ;
            ;4895 ; Either/Or Bits 28 and 9 were not set in configuration register C/D.
            ;4896 ;
            ;4897 =0
            ;4898 T.43.WRONG.BITS.SET.1:
U 11AC, 0000.003D.D580.0800.0084.709E   ;4899  ERROR2[.6]
U 11AD, 0000.003C.0180.0800.0000.11AE   ;4900  NOP
            ;4901 ;
            ;4902 ;////////////////////////////////////////////////////////////////////////////////
            ;4903 ;
            ;4904 ; ERROR LOGOUT:
            ;4905 ;
            ;4906 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4907 ;  I/O Address of Configuration Register C/D
            ;4908 ;  Unused
            ;4909 ;  Configuration Register C/D Bits that are Under Test
            ;4910 ;  Configuration Register C/D Data
            ;4911 ;  Unused
            ;4912 ;
            ;4913 ;////////////////////////////////////////////////////////////////////////////////
            ;4914 ;
            ;4915 =0
            ;4916 T.43.BITS.OK.1:
U 11AE, 0000.003D.0180.0800.0000.1228   ;4917  ERLOOP    ; loop on error from here
U 11AF, 0000.003C.0180.0800.0000.11B0   ;4918  NOP
U 11B0, 0000.003D.0180.0800.0000.10AC   ;4919 =0 CALL[SBI.INIT]  ; initialize the sbi
U 11B1, 0000.003C.0180.0800.0000.11B2   ;4920  NOP
U 11B2, 0000.003D.0180.0800.0000.12B5   ;4921 =0 CALL[MS780E.CLEANUP] ; Clear memory configuration registers
            ;4922  Q_K[.1],  ; Data for error to be forced in upper longword
U 11B3, 0F18.0038.05C0.0800.0000.11B4   ;4923   D_0  ; No error in lower longword
U 11B4, 0000.003D.0180.0800.0000.12D5   ;4924 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11B5, 001B.0010.8980.0800.0082.1074   ;4925  SC_K[.D]+1  ; Load SC with x0E
U 1074, 0000.003D.0180.0800.0000.12C5   ;4926 =00 CALL[CHECK_SBI_ERR] ; Is the CRD bit set in the SBI error register?
U 1075, 0000.003C.0180.0800.0000.1126   ;4927  J/T.43.CRD.NOT.SET.1 ; No, call an error
U 1076, 0000.003C.0180.0800.0000.1078   ;4928  J/T.43.CRD.SET.1 ; yes, continue with the test
            ;4929 =
```

```
            ;4930  .PAGE
            ;4931 ;
            ;4932 ;////////////////////////////////////////////////////////////////////////////////
            ;4933 ;
            ;4934 ; SBI Error register bit 14 (CRD) was not set and should have been.
            ;4935 ; Call an error.
            ;4936 ;
            ;4937 =0
            ;4938 T.43.CRD.NOT.SET.1:
U 11B6, 0000,003D,5D80,0800,0084,709E  ;4939  ERROR2[.7]
U 11B7, 0000,003C,0180,0800,0000,1078  ;4940  NOP
            ;4941 ;
            ;4942 ;////////////////////////////////////////////////////////////////////////////////
            ;4943 ;
            ;4944 ; ERROR LOGOUT:
            ;4945 ;
            ;4946 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4947 ;  I/O Address of Configuration Register C/D
            ;4948 ;  Unused
            ;4949 ;  Configuration Register C/D Bits that are Under Test
            ;4950 ;  Configuration Register C/D Data
            ;4951 ;  Unused
            ;4952 ;  ID[sbi.err] Register
            ;4953 ;
            ;4954 ;////////////////////////////////////////////////////////////////////////////////
            ;4955 ;
            ;4956 =00
            ;4957 T.43.CRD.SET.1:
U 1078, 0000,003D,8980,0800,0084,72C5  ;4958  CHECK.SBI.ERR[.D] ; Is the RDS bit set in SBI Error ?
U 1079, 0000,003C,0180,0800,0000,11B8  ;4959  J/T.43.DO.NEXT.1 ; No, continue with the test
```

```
            ;4960  .PAGE
            ;4961 ;
            ;4962 ;////////////////////////////////////////////////////////////////////////////////
            ;4963 ;
            ;4964 ; SBI Error Register bit 13 (RDS) was set and should not have been.
            ;4965 ; Call an error.
            ;4966 ;
U 107A, 0000,003D,5D80,0800,0084,709E  ;4967  ERROR2[.7]
U 107B, 0000,003C,0180,0800,0000,11B8  ;4968  NOP
            ;4969 ;
            ;4970 ;////////////////////////////////////////////////////////////////////////////////
            ;4971 ;
            ;4972 ; ERROR LOGOUT:
            ;4973 ;
            ;4974 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4975 ;  I/O Address of Configuration Register C/D
            ;4976 ;  Unused
            ;4977 ;  Configuration Register C/D Bits that are Under Test
            ;4978 ;  Configuration Register C/D Data
            ;4979 ;  Unused
            ;4980 ;  ID[sbi.err] Register
            ;4981 ;
            ;4982 ;////////////////////////////////////////////////////////////////////////////////
            ;4983 ;
            ;4984 =0
            ;4985 T.43.DO.NEXT.1:
U 11B8, 0000,003D,0180,0800,0000,10AC  ;4986  CALL[SBI.INIT]  ; Initialize the SBI
U 11B9, 0018,0038,1180,0800,0200,13B4  ;4987  VA_K[.4]  ; Point VA to the second longword
U 13B4, 0000,003C,0180,C800,0000,13B5  ;4988  D[LONG]_CACHE.P  ; Read the second longword
U 13B5, 0001,003C,0180,0800,0010,13B6  ;4989  ALU_D,CLK.UBCC  ; Move second longword to ALU for validity chk
U 13B6, 0000,013C,0180,0800,0000,11BA  ;4990  Z?  ; Is the value read equal to 0 ?
U 11BA, 0000,003C,0180,0800,0000,11BC  ;4991 =0 J/T.43.BAD.DATA.1 ; No, Call an error
U 11BB, 0000,003C,0180,0800,0000,11BE  ;4992  J/T.43.S2  ; Yes, continue with the test
```

```
            ;4993  .PAGE
            ;4994 ;
            ;4995 ;////////////////////////////////////////////////////////////////////////////
            ;4996 ;
            ;4997 ; Data read from the second longword was not equal to zero.  Call an error.
            ;4998 ;
            ;4999 =0
            ;5000 T.43.BAD.DATA.1:
U 11BC, 0000.003D.D580.0800.0084.709E  ;5001  ERROR2[.6]
U 11BD, 0000.003C.0180.0800.0000.11BE  ;5002  NOP
            ;5003 ;
            ;5004 ;////////////////////////////////////////////////////////////////////////////
            ;5005 ;
            ;5006 ; ERROR LOGOUT:
            ;5007 ;
            ;5008 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5009 ;  I/O Address of Configuration Register C/D
            ;5010 ;  Unused
            ;5011 ;  Configuration Register C/D Bits that are Under Test
            ;5012 ;  Configuration Register C/D Data
            ;5013 ;  Unused
            ;5014 ;
            ;5015 ;////////////////////////////////////////////////////////////////////////////
            ;5016 ;
            ;5017 ;
            ;5018 ;============================================================================
            ;5019 ;
            ;5020 ; Subtest 2
            ;5021 ;
            ;5022 =0
            ;5023 T.43.S2:
U 11BE, 0000.003D.0180.0800.0000.1230  ;5024  SUBTEST   ; Update the Subtest counter
U 11BF, 001B.0010.ED80.0800.0082.13B7  ;5025  SC_K[.1B]+1  ; Load SC with x1C to mask bit 28
U 13B7, 0803.001C.0180.0800.0000.13B8  ;5026  D_NOT.MASK  ; Assert bit 28 in the D register
U 13B8, 0000.003C.F580.0800.0084.73B9  ;5027  SC_K[.A]  ; Get ready to mask bit 10
U 13B9, 0801.001C.0180.0800.0000.13BA  ;5028  D_D.ORNOT.MASK  ; Assert bits 28 & bit 10 in the D register
U 13BA, 0001.003C.0180.0A88.0000.11C0  ;5029  R[01]_D   ; Load R[01] with x10000400 which is the
            ;5030      ; expected set bits in CNFG C/D
U 11C0, 0000.003D.0180.0800.0000.1228  ;5031 =0 ERLOOP   ; loop on error from here
U 11C1, 0000.003C.0180.0800.0000.11C2  ;5032  NOP
U 11C2, 0000.003D.0180.0800.0000.10AC  ;5033 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 11C3, 0000.003C.0180.0800.0000.11C4  ;5034  NOP
U 11C4, 0000.003D.0180.0800.0000.12B5  ;5035 =0 CALL[MS780E.CLEANUP] ; Clear the memory configuration registers
U 11C5, 0018.0038.0DC0.0800.0000.13BB  ;5036  Q_K[.3]   ; Data for error to be forced in the upper
            ;5037      ; longword (RDS in this case)
U 13BB, 0F00.003C.0180.0800.0000.11C6  ;5038  D_0   ; No error forced in lower longword
U 11C6, 0000.003D.0180.0800.0000.12D5  ;5039 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11C7, 0000.003C.0180.0800.0000.107C  ;5040  NOP
U 107C, 0000.003D.0180.0800.0000.1265  ;5041 =00 CHECK.UTRAP  ; Was there a RDS micro trap ?
U 107D, 0000.003C.0180.0800.0000.107F  ;5042  J/T.43.NO.RDS.UTRAP.2 ; No, continue with the test
```

```
            ;5043  .PAGE
            ;5044 ;
            ;5045 ;////////////////////////////////////////////////////////////////////////////////
            ;5046 ;
            ;5047 ; Got a RDS micro trap when forcing the CRD error.  Call an error.
            ;5048 ;
U 107E, 0000,003D,0180,0800,0084,709E  ;5049  ERROR2[.8]
            ;5050 ;
            ;5051 ;////////////////////////////////////////////////////////////////////////////////
            ;5052 ;
            ;5053 ; ERROR LOGOUT:
            ;5054 ;
            ;5055 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5056 ;  I/O Address of Configuration Register C/D
            ;5057 ;  Unused
            ;5058 ;  Configuration Register C/D Bits that are Under Test
            ;5059 ;  Configuration Register C/D Data
            ;5060 ;  Unused
            ;5061 ;  Expected Micro Trap Flags
            ;5062 ;  Received Micro Trap Flags
            ;5063 ;
            ;5064 ;////////////////////////////////////////////////////////////////////////////////
            ;5065 ;
            ;5066 T.43.NO.RDS.UTRAP.2:
U 107F, 0010,0038,0180,0968,0200,13BC  ;5067  VA_RC[0D]  ; Point VA to Configuration register c/d
U 13BC, 0000,003C,0180,C800,0000,13BD  ;5068  D[LONG]_CACHE.P  ; read configuration register c/d and
U 13BD, 0001,003C,0180,09D0,0000,13BE  ;5069  RC[0A]_D  ; store in RC[0A]
U 13BE, 081C,2034,0180,0A00,0000,13BF  ;5070  D_D.AND.R[0]  ; Mask out the unwanted bits and then
U 13BF, 0001,003C,0180,09D8,0000,13C0  ;5071  RC[0B]_D  ; store in RC[0B]
            ;5072  ALU_D.XOR.R[1],  ; Exclusive Or expected error bits with the
U 13C0, 000D,0020,0180,0A08,0010,13C1  ;5073  CLK.UBCC  ; received error bits
U 13C1, 0000,013C,0180,0800,0000,11C8  ;5074  Z?  ; Were the correct error bits set ?
```

```
            ;5075  .PAGE
            ;5076 ;
            ;5077 ;////////////////////////////////////////////////////////////////////////////
            ;5078 ;
            ;5079 ; Either/Or Bits 28 and 10 were not set in configuration register C/D.
            ;5080 ;
            ;5081 =0
            ;5082 T.43.WRONG.BITS.SET.2:
U 11C8. 0000.003D.D580.0800.0084.709E  ;5083  ERROR2[.6]
U 11C9. 0000.003C.0180.0800.0000.11CA  ;5084  NOP
            ;5085 ;
            ;5086 ;////////////////////////////////////////////////////////////////////////////
            ;5087 ;
            ;5088 ; ERROR LOGOUT:
            ;5089 ;
            ;5090 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5091 ;  I/O Address of Configuration Register C/D
            ;5092 ;  Unused
            ;5093 ;  Configuration Register C/D Bits that are Under Test
            ;5094 ;  Configuration Register C/D Data
            ;5095 ;  Unused
            ;5096 ;
            ;5097 ;////////////////////////////////////////////////////////////////////////////
            ;5098 ;
            ;5099 =0
            ;5100 T.43.BITS.OK.2:
U 11CA. 0000.003D.0180.0800.0000.1228  ;5101  ERLOOP   ; loop on error from here
U 11CB. 0000.003C.0180.0800.0000.11CC  ;5102  NOP
U 11CC. 0000.003D.0180.0800.0000.10AC  ;5103 =0 CALL[SBI.INIT]  ; initialize the sbi
U 11CD. 0000.003C.0180.0800.0000.11CE  ;5104  NOP
U 11CE. 0000.003D.0180.0800.0000.12B5  ;5105 =0 CALL[MS780E.CLEANUP] ; Clear memory configuration registers
U 11CF. 0018.0038.0DC0.0800.0000.13C2  ;5106  Q_K[.3]   ; Data for error to be forced in upper longword
U 13C2. 0F00.003C.0180.0800.0000.11D0  ;5107  D_0   ; No error in lower longword
U 11D0. 0000.003D.0180.0800.0000.12D5  ;5108 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11D1. 0000.003C.0180.0800.0000.1080  ;5109  NOP
U 1080. 0000.003D.8980.0800.0084.72C5  ;5110 =00 CHECK.SBI.ERR[.D] ; Did the RDS bit in SBI Error reg. get set ?
U 1081. 0000.003C.0180.0800.0000.11D2  ;5111  J/T.43.RDS.NOT.SET.1 ; No, call an error
U 1082. 0000.003C.0180.0800.0000.11D3  ;5112  J/T.43.RDS.IS.SET.1 ; Yes, continue with the test
            ;5113 =
```

```
            ;5114  .PAGE
            ;5115 ;
            ;5116 ;////////////////////////////////////////////////////////////////////////////////
            ;5117 ;
            ;5118 ; SBI Error register bit 13 (RDS) did not set and should have.  Call an Error.
            ;5119 ;
            ;5120 =0
            ;5121 T.43.RDS.NOT.SET.1:
U 11D2, 0000,003D,5D80,0800,0084,709E  ;5122  ERROR2[.7]
            ;5123 ;
            ;5124 ;////////////////////////////////////////////////////////////////////////////////
            ;5125 ;
            ;5126 ; ERROR LOGOUT:
            ;5127 ;
            ;5128 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5129 ;  I/O Address of Configuration Register C/D
            ;5130 ;  Unused
            ;5131 ;  Configuration Register C/D Bits that are Under Test
            ;5132 ;  Configuration Register C/D Data
            ;5133 ;  Unused
            ;5134 ;  ID[sbi.err] Register
            ;5135 ;
            ;5136 ;////////////////////////////////////////////////////////////////////////////////
            ;5137 ;
            ;5138 T.43.RDS.IS.SET.1:
U 11D3, 001B,0010,8980,0800,0082,1084  ;5139  SC_K[.D]+1  ; Load SC with x0E
U 1084, 0000,003D,0180,0800,0000,12C5  ;5140 =00 CALL[CHECK_SBI_ERR] ; Is the CRD bit set in the SBI error register?
U 1085, 0000,003C,0180,0800,0000,11D4  ;5141  J/T.43.CRD.NOT.SET.2 ; OK. CRD Bit NOT set
```

```
            ;5142  .PAGE
            ;5143 ;
            ;5144 ;////////////////////////////////////////////////////////////////////////////////
            ;5145 ;
            ;5146 ; SBI Error register bit 14 (CRD) was set and shouldn't have been.
            ;5147 ; Call an error.
            ;5148 ;
U 1086, 0000,003D,5D80,0800,0084,709E  ;5149  ERROR2[.7]
U 1087, 0000,003C,0180,0800,0000,11D4  ;5150  NOP
            ;5151 ;
            ;5152 ;////////////////////////////////////////////////////////////////////////////////
            ;5153 ;
            ;5154 ; ERROR LOGOUT:
            ;5155 ;
            ;5156 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5157 ;  I/O Address of Configuration Register C/D
            ;5158 ;  Unused
            ;5159 ;  Configuration Register C/D Bits that are Under Test
            ;5160 ;  Configuration Register C/D Data
            ;5161 ;  Unused
            ;5162 ;  ID[sbi.err] Register
            ;5163 ;
            ;5164 ;////////////////////////////////////////////////////////////////////////////////
            ;5165 ;
            ;5166 =0
            ;5167 T.43.CRD.NOT.SET.2:
U 11D4, 0000,003D,0180,0800,0000,10AC  ;5168  CALL[SBI.INIT]  ; initialize the SBI
U 11D5, 0018,0038,1180,0800,0200,13C3  ;5169  VA_K[.4]  ; Point VA to the second longword
U 13C3, 0000,003C,0180,C800,0000,13C4  ;5170  D[LONG]_CACHE.P  ; Read the second longword
            ;5171  ALU_D.XOR.K[.3], ; Exclusive the contents of second longword
U 13C4, 0019,0020,0D80,0800,0010,13C5  ;5172  CLK.UBCC  ; with x3
U 13C5, 0000,013C,0180,0800,0000,11D6  ;5173  Z?  ; Was the second longword equal to x3 ?
```

```
            ;5174  .PAGE
            ;5175 ;
            ;5176 ;////////////////////////////////////////////////////////////////////////////////
            ;5177 ;
            ;5178 ; Second longword did not match expected data.  Call an error.
            ;5179 ;
            ;5180 =0
            ;5181 T.43.BAD.DATA.2:
U 11D6. 0000.003D.D580.0800.0084.709E  ;5182  ERROR2[.6]
U 11D7. 0000.003C.0180.0800.0000.11D8  ;5183  NOP
            ;5184 ;
            ;5185 ;////////////////////////////////////////////////////////////////////////////////
            ;5186 ;
            ;5187 ; ERROR LOGOUT:
            ;5188 ;
            ;5189 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5190 ;  I/O Address of Configuration Register C/D
            ;5191 ;  Unused
            ;5192 ;  Configuration Register C/D Bits that are Under Test
            ;5193 ;  Configuration Register C/D Data
            ;5194 ;  Unused
            ;5195 ;
            ;5196 ;////////////////////////////////////////////////////////////////////////////////
            ;5197 ;
            ;5198 ;
            ;5199 ;================================================================================
            ;5200 ;
            ;5201 ; Subtest 3
            ;5202 ;
            ;5203 =0
            ;5204 T.43.S3:
U 11D8. 0000.003D.0180.0800.0000.1230  ;5205  SUBTEST   ; Update the subtest counter
U 11D9. 0000.003C.0180.0800.0000.11DA  ;5206  NOP
U 11DA. 0000.003D.0180.0800.0000.1228  ;5207 =0 ERLOOP   ; loop on error from here
U 11DB. 0000.003C.0180.0800.0000.11DC  ;5208  NOP
U 11DC. 0000.003D.0180.0800.0000.10AC  ;5209 =0 CALL[SBI.INIT]  ; initialize the sbi
U 11DD. 0000.003C.0180.0800.0000.11DE  ;5210  NOP
U 11DE. 0000.003D.0180.0800.0000.12B5  ;5211 =0 CALL[MS780E.CLEANUP] ; clear memory configuration registers
U 11DF. 0018.0038.0DC0.0800.0000.13C6  ;5212  Q_K[.3]   ; Data for error to be forced in upper longword
            ;5213      ; this will be a RDS
U 13C6. 0818.0038.0580.0800.0000.11E0  ;5214  D_K[.1]   ; Data for error to be forced in lower longword
            ;5215      ; this will be a CRD
U 11E0. 0000.003D.0180.0800.0000.12D5  ;5216 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11E1. 0000.003C.0180.0800.0000.1088  ;5217  NOP
U 1088. 0000.003D.0180.0800.0000.1265  ;5218 =00 CHECK.UTRAP  ; was there a RDS micro trap ?
U 1089. 0000.003C.0180.0800.0000.108B  ;5219  J/T.43.NO.RDS.UTRAP.3 ; No, continue with the test
```

```
            ;5220  .PAGE
            ;5221 ;
            ;5222 ;////////////////////////////////////////////////////////////////////////////////
            ;5223 ;
            ;5224 ; Got a RDS micro trap and shouldn't have.  Call an error.
            ;5225 ;
U 108A. 0000.003D.0180.0800.0084.709E  ;5226  ERROR2[.8]
            ;5227 ;
            ;5228 ;////////////////////////////////////////////////////////////////////////////////
            ;5229 ;
            ;5230 ; ERROR LOGOUT:
            ;5231 ;
            ;5232 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5233 ;  I/O Address of Configuration Register C/D
            ;5234 ;  Unused
            ;5235 ;  Configuration Register C/D Bits that are Under Test
            ;5236 ;  Configuration Register C/D Data
            ;5237 ;  Unused
            ;5238 ;  Expected Micro Trap Flags
            ;5239 ;  Received Micro Trap Flags
            ;5240 ;
            ;5241 ;////////////////////////////////////////////////////////////////////////////////
            ;5242 ;
            ;5243 T.43.NO.RDS.UTRAP.3:
U 108B. 0010.0038.0180.0968.0200.13C7  ;5244  VA_RC[0D]  ; point VA to configuration register c/d
U 13C7. 0000.003C.0180.C800.0000.13C8  ;5245  D[LONG]_CACHE.P  ; Read configuration register c/d
U 13C8. 0001.003C.0180.09D0.0000.13C9  ;5246  RC[0A]_D  ; save cnfg c/d data in RC[0A]
U 13C9. 081C.2034.0180.0A00.0000.13CA  ;5247  D_D.AND.R[0]  ; mask out unwanted bits
U 13CA. 0001.003C.0180.09D8.0000.13CB  ;5248  RC[0B]_D  ; and save in RC[0B]
            ;5249  ALU_D.XOR.R[0],  ; Exclusive Or expected error bits to be set
U 13CB. 000D.0020.0180.0A00.0010.13CC  ;5250  CLK.UBCC  ; with received error bits
U 13CC. 0000.013C.0180.0800.0000.11E2  ;5251  Z?  ; were the bits set ?
```

```
            ;5252  .PAGE
            ;5253 ;
            ;5254 ;////////////////////////////////////////////////////////////////////////////
            ;5255 ;
            ;5256 ; Either/Or Bits 28:29 and 10:9 were incorrect in Configuration register C/D.
            ;5257 ;
            ;5258 =0
            ;5259 T.43.WRONG.BITS.SET.3:
U 11E2. 0000.003D.D580.0800.0084.709E  ;5260  ERROR2[.6]
U 11E3. 0000.003C.0180.0800.0000.11E4  ;5261  NOP
            ;5262 ;
            ;5263 ;////////////////////////////////////////////////////////////////////////////
            ;5264 ;
            ;5265 ; ERROR LOGOUT:
            ;5266 ;
            ;5267 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5268 ;  I/O Address of Configuration Register C/D
            ;5269 ;  Unused
            ;5270 ;  Configuration Register C/D Bits that are Under Test
            ;5271 ;  Configuration Register C/D Data
            ;5272 ;  Unused
            ;5273 ;
            ;5274 ;////////////////////////////////////////////////////////////////////////////
            ;5275 ;
            ;5276 =0
            ;5277 T.43.BITS.OK.3:
U 11E4. 0000.003D.0180.0800.0000.1228  ;5278  ERLOOP   ; loop on error from here
U 11E5. 0000.003C.0180.0800.0000.11E6  ;5279  NOP
U 11E6. 0000.003D.0180.0800.0000.10AC  ;5280 =0 CALL[SBI.INIT]  ; initialize the sbi
U 11E7. 0000.003C.0180.0800.0000.11E8  ;5281  NOP
U 11E8. 0000.003D.0180.0800.0000.12B5  ;5282 =0 CALL[MS780E.CLEANUP] ; Clear memory configuration registers
U 11E9. 0018.0038.0DC0.0800.0000.13CD  ;5283  Q_K[.3]   ; Data for error to be forced in upper longword
            ;5284      ; this will be a RDS error
U 13CD. 0818.0038.0580.0800.0000.11EA  ;5285  D_K[.1]   ; Data for error to be forced in lower longword
            ;5286      ; this will be a CRD error
U 11EA. 0000.003D.0180.0800.0000.12D5  ;5287 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11EB. 0000.003C.0180.0800.0000.108C  ;5288  NOP
U 108C. 0000.003D.8980.0800.0084.72C5  ;5289 =00 CHECK.SBI.ERR[.D] ; Was the RDS bit set in SBI Error reg. ?
U 108D. 0000.003C.0180.0800.0000.11EC  ;5290  J/T.43.RDS.NOT.SET.2 ; No, call an error
U 108E. 0000.003C.0180.0800.0000.11ED  ;5291  J/T.43.RDS.SET.2 ; Yes, continue with the test
            ;5292 =
```

```
            ;5293  .PAGE
            ;5294 ;
            ;5295 ;////////////////////////////////////////////////////////////////////////////////
            ;5296 ;
            ;5297 ; SBI Error register bit 13 (RDS) did not set.  Call an error.
            ;5298 ;
            ;5299 =0
            ;5300 T.43.RDS.NOT.SET.2:
U 11EC, 0000,003D,5D80,0800,0084,709E  ;5301  ERROR2[.7]
            ;5302 ;
            ;5303 ;////////////////////////////////////////////////////////////////////////////////
            ;5304 ;
            ;5305 ; ERROR LOGOUT:
            ;5306 ;
            ;5307 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5308 ;  I/O Address of Configuration Register C/D
            ;5309 ;  Unused
            ;5310 ;  Configuration Register C/D Bits that are Under Test
            ;5311 ;  Configuration Register C/D Data
            ;5312 ;  Unused
            ;5313 ;  ID[sbi.err] Register
            ;5314 ;
            ;5315 ;////////////////////////////////////////////////////////////////////////////////
            ;5316 ;
            ;5317 T.43.RDS.SET.2:
U 11ED, 001B,0010,8980,0800,0082,1090  ;5318  SC_K[.D]+1  ; Load SC with xE
U 1090, 0000,003D,0180,0800,0000,12C5  ;5319 =00 CALL[CHECK_SBI_ERR] ; Was the CRD bit set in SBI Error ?
U 1091, 0000,003C,0180,0800,0000,11EE  ;5320  J/T.43.CRD.NOT.SET.3 ; No, call an error
U 1092, 0000,003C,0180,0800,0000,11F0  ;5321  J/T.43.CRD.SET.3 ; Yes, continue with the test
            ;5322 =
```

```
            ;5323  .PAGE
            ;5324 ;
            ;5325 ;////////////////////////////////////////////////////////////////////////////////
            ;5326 ;
            ;5327 ; SBI Error Register bit 14 (CRD) did not set.  Call an error.
            ;5328 ;
            ;5329 =0
            ;5330 T.43.CRD.NOT.SET.3:
U 11EE, 0000.003D.5D80.0800.0084.709E  ;5331  ERROR2[.7]
U 11EF, 0000.003C.0180.0800.0000.11F0  ;5332  NOP
            ;5333 ;
            ;5334 ;////////////////////////////////////////////////////////////////////////////////
            ;5335 ;
            ;5336 ; ERROR LOGOUT:
            ;5337 ;
            ;5338 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5339 ;  I/O Address of Configuration Register C/D
            ;5340 ;  Unused
            ;5341 ;  Configuration Register C/D Bits that are Under Test
            ;5342 ;  Configuration Register C/D Data
            ;5343 ;  Unused
            ;5344 ;  ID[sbi.err] Register
            ;5345 ;
            ;5346 ;////////////////////////////////////////////////////////////////////////////////
            ;5347 ;
            ;5348 =0
            ;5349 T.43.CRD.SET.3:
U 11F0, 0000.003D.0180.0800.0000.10AC  ;5350  CALL[SBI.INIT]  ; initialize the sbi
U 11F1, 0018.0038.1180.0800.0200.13CE  ;5351  VA_K[.4]  ; Point VA to the second longword
U 13CE, 0000.003C.0180.C800.0000.13CF  ;5352  D[LONG]_CACHE.P  ; Read the second longword
            ;5353  ALU_D.XOR.K[.3], ; Exclusive Or the received data with the
U 13CF, 0019.0020.0D80.0800.0010.13D0  ;5354  CLK.UBCC  ; expected data
U 13D0, 0000.013C.0180.0800.0000.11F2  ;5355  Z?  ; Are they equal ?
```

```
            ;5356  .PAGE
            ;5357 ;
            ;5358 ;////////////////////////////////////////////////////////////////////////////
            ;5359 ;
            ;5360 ; Received data from second longword did not match expected data.
            ;5361 ; Call an error.
            ;5362 ;
            ;5363 =0
            ;5364 T.43.BAD.DATA.3:
U 11F2, 0000.003D.D580.0800.0084.709E  ;5365  ERROR2[.6]
U 11F3, 0000.003C.0180.0800.0000.11F4  ;5366  NOP
            ;5367 ;
            ;5368 ;////////////////////////////////////////////////////////////////////////////
            ;5369 ;
            ;5370 ; ERROR LOGOUT:
            ;5371 ;
            ;5372 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5373 ;  I/O Address of Configuration Register C/D
            ;5374 ;  Unused
            ;5375 ;  Configuration Register C/D Bits that are Under Test
            ;5376 ;  Configuration Register C/D Data
            ;5377 ;  Unused
            ;5378 ;
            ;5379 ;////////////////////////////////////////////////////////////////////////////
            ;5380 ;
            ;5381 ;
            ;5382 ;============================================================================
            ;5383 ;
            ;5384 ; Subtest 4
            ;5385 ;
            ;5386 =0
            ;5387 T.43.S4:
U 11F4, 0000.003D.0180.0800.0000.1230  ;5388  SUBTEST   ; Update the subtest counter
U 11F5, 0000.003C.0180.0800.0000.11F6  ;5389  NOP
U 11F6, 0000.003D.0180.0800.0000.1228  ;5390 =0 ERLOOP   ; loop on error from here
U 11F7, 0000.003C.0180.0800.0000.11F8  ;5391  NOP
U 11F8, 0000.003D.0180.0800.0000.10AC  ;5392 =0 CALL[SBI.INIT]  ; initialize the sbi
U 11F9, 0000.003C.0180.0800.0000.11FA  ;5393  NOP
U 11FA, 0000.003D.0180.0800.0000.12B5  ;5394 =0 CALL[MS780E.CLEANUP] ; clear memory configuration registers
U 11FB, 0018.0038.05C0.0800.0000.13D1  ;5395  Q_K[.1]   ; Data for error to be forced in upper longword
            ;5396      ; this will be a CRD
U 13D1, 0818.0038.0D80.0800.0000.11FC  ;5397  D_K[.3]   ; Data for error to be forced in lower longword
            ;5398      ; this will be a RDS
U 11FC, 0000.003D.0180.0800.0000.12D5  ;5399 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 11FD, 0000.003C.0180.0800.0000.1094  ;5400  NOP
U 1094, 0000.003D.0180.0800.0000.1265  ;5401 =00 CHECK.UTRAP  ; was there a RDS micro trap ?
U 1095, 0000.003C.0180.0800.0000.11FE  ;5402  J/T.43.NO.RDS.UTRAP.4 ; No, call an error
U 1096, 0000.003C.0180.0800.0000.11FF  ;5403  J/T.43.RDS.UTRAP.3 ; Yes, continue with the test
            ;5404 =
```

```
            ;5405  .PAGE
            ;5406 ;
            ;5407 ;////////////////////////////////////////////////////////////////////////////////
            ;5408 ;
            ;5409 ; Got a RDS micro trap and shouldn't have.  Call an error.
            ;5410 ;
            ;5411 =0
            ;5412 T.43.NO.RDS.UTRAP.4:
U 11FE, 0000,003D,0180,0800,0084,709E  ;5413  ERROR2[.8]
            ;5414 ;
            ;5415 ;////////////////////////////////////////////////////////////////////////////////
            ;5416 ;
            ;5417 ; ERROR LOGOUT:
            ;5418 ;
            ;5419 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5420 ;  I/O Address of Configuration Register C/D
            ;5421 ;  Unused
            ;5422 ;  Configuration Register C/D Bits that are Under Test
            ;5423 ;  Configuration Register C/D Data
            ;5424 ;  Unused
            ;5425 ;  Expected Micro Trap Flags
            ;5426 ;  Received Micro Trap Flags
            ;5427 ;
            ;5428 ;////////////////////////////////////////////////////////////////////////////////
            ;5429 ;
            ;5430 T.43.RDS.UTRAP.3:
U 11FF, 0010,0038,0180,0968,0200,13D2  ;5431  VA_RC[0D]  ; point VA to configuration register c/d
U 13D2, 0000,003C,0180,C800,0000,13D3  ;5432  D[LONG]_CACHE.P  ; Read configuration register c/d
U 13D3, 0001,003C,0180,09D0,0000,13D4  ;5433  RC[0A]_D  ; save cnfg c/d data in RC[0A]
U 13D4, 081C,2034,0180,0A00,0000,13D5  ;5434  D_D.AND.R[0]  ; mask out unwanted bits
U 13D5, 0001,003C,0180,09D8,0000,13D6  ;5435  RC[0B]_D  ; and save in RC[0B]
            ;5436  ALU_D.XOR.R[0],  ; Exclusive Or expected error bits to be set
U 13D6, 000D,0020,0180,0A00,0010,13D7  ;5437  CLK.UBCC  ; with received error bits
U 13D7, 0000,013C,0180,0800,0000,1200  ;5438  Z?  ; were the bits set ?
```

```
            ;5439  .PAGE
            ;5440 :
            ;5441 ;////////////////////////////////////////////////////////////////////////////////
            ;5442 ;
            ;5443 ; Either/Or Bits 29:28 and 10:9 were incorrect in configuration register C/D.
            ;5444 ;
            ;5445 =0
            ;5446 T.43.WRONG.BITS.SET.4:
U 1200, 0000,003D,D580,0800,0084,709E  ;5447  ERROR2[.6]
U 1201, 0000,003C,0180,0800,0000,1202  ;5448  NOP
            ;5449 ;
            ;5450 ;////////////////////////////////////////////////////////////////////////////////
            ;5451 ;
            ;5452 ; ERROR LOGOUT:
            ;5453 ;
            ;5454 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5455 ;  I/O Address of Configuration Register C/D
            ;5456 ;  Unused
            ;5457 ;  Configuration Register C/D Bits that are Under Test
            ;5458 ;  Configuration Register C/D Data
            ;5459 ;  Unused
            ;5460 ;
            ;5461 ;////////////////////////////////////////////////////////////////////////////////
            ;5462 ;
            ;5463 =0
            ;5464 T.43.BITS.OK.4:
U 1202, 0000,003D,0180,0800,0000,1228  ;5465  ERLOOP   ; loop on error from here
U 1203, 0000,003C,0180,0800,0000,1204  ;5466  NOP
U 1204, 0000,003D,0180,0800,0000,10AC  ;5467 =0 CALL[SBI.INIT]  ; initialize the sbi
U 1205, 0000,003C,0180,0800,0000,1206  ;5468  NOP
U 1206, 0000,003D,0180,0800,0000,12B5  ;5469 =0 CALL[MS780E.CLEANUP] ; Clear memory configuration registers
U 1207, 0018,0038,05C0,0800,0000,13D8  ;5470  Q_K[.1]   ; Data for error to be forced in upper longword
        ;5471     ; this will be a CRD
U 13D8, 0818,0038,0D80,0800,0000,1208  ;5472  D_K[.3]   ; Data for error to be forced in lower longword
        ;5473     ; this will be a RDS
U 1208, 0000,003D,0180,0800,0000,12D5  ;5474 =0 CALL[FORCE.CRD.RDS.READ]; Force the errors and read the data
U 1209, 001B,0010,8980,0800,0082,1098  ;5475  SC_K[.D]+1  ; Load SC with xE
U 1098, 0000,003D,0180,0800,0000,12C5  ;5476 =00 CALL[CHECK_SBI_ERR] ; Was the CRD bit set in SBI Error ?
U 1099, 0000,003C,0180,0800,0000,120A  ;5477  J/T.43.CRD.NOT.SET.4 ; No, call an error
```

```
            ;5478  .PAGE
            ;5479 ;
            ;5480 ;////////////////////////////////////////////////////////////////////////////////
            ;5481 ;
            ;5482 ; SBI Error Register bit 14 (CRD) did not set.  Call an error.
            ;5483 ;
            ;5484 =0
            ;5485 T.43.CRD.SET.4:
U 109A. 0000.003D.5D80.0800.0084.709E  ;5486  ERROR2[.7]
U 109B. 0000.003C.0180.0800.0000.120A  ;5487  NOP
            ;5488 ;
            ;5489 ;////////////////////////////////////////////////////////////////////////////////
            ;5490 ;
            ;5491 ; ERROR LOGOUT:
            ;5492 ;
            ;5493 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5494 ;  I/O Address of Configuration Register C/D
            ;5495 ;  Unused
            ;5496 ;  Configuration Register C/D Bits that are Under Test
            ;5497 ;  Configuration Register C/D Data
            ;5498 ;  Unused
            ;5499 ;  ID[sbi.err] Register
            ;5500 ;
            ;5501 ;////////////////////////////////////////////////////////////////////////////////
            ;5502 ;
            ;5503 =0
            ;5504 T.43.CRD.NOT.SET.4:
U 120A. 0000.003D.0180.0800.0000.10AC  ;5505  CALL[SBI.INIT]  ; initialize the sbi
U 120B. 0018.0038.1180.0800.0200.13D9  ;5506  VA_K[.4]  ; Point VA to the second longword
U 13D9. 0000.003C.0180.C800.0000.13DA  ;5507  D[LONG]_CACHE.P  ; Read the second longword
            ;5508  ALU_D,   ; Load ALU with the received data
U 13DA. 0001.003C.0180.0800.0010.13DB  ;5509  CLK.UBCC  ; ...
U 13DB. 0000.013C.0180.0800.0000.120C  ;5510  Z?   ; is it equal to zero ?
```

```
            ;5511   .PAGE
            ;5512 ;
            ;5513 ;////////////////////////////////////////////////////////////////////////////////
            ;5514 ;
            ;5515 ; Received data from second longword did not match expected data.
            ;5516 ; Call an error.
            ;5517 ;
            ;5518 =0
            ;5519 T.43.BAD.DATA.4:
U 120C, 0000,003D,D580,0800,0084,709E   ;5520   ERROR2[.6]
U 120D, 0000,003C,0180,0800,0000,106C   ;5521   J/RESTART.TEST.43
            ;5522 ;
            ;5523 ;////////////////////////////////////////////////////////////////////////////////
            ;5524 ;
            ;5525 ; ERROR LOGOUT:
            ;5526 ;
            ;5527 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;5528 ;  I/O Address of Configuration Register C/D
            ;5529 ;  Unused
            ;5530 ;  Configuration Register C/D Bits that are Under Test
            ;5531 ;  Configuration Register C/D Data
            ;5532 ;  Unused
            ;5533 ;
            ;5534 ;////////////////////////////////////////////////////////////////////////////////
            ;5535 ;
            ;5536  ;
            ;5537  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;5538  ;
            ;5539 =0
            ;5540 T.43.EXIT:
U 120E, 0000,003D,0180,0800,0000,12B5   ;5541   CALL[MS780E.CLEANUP]  ; Clear Memory CNFG Regs before exitting
U 120F, 0000,003C,0180,0800,0000,1210   ;5542   NOP
            ;5543
            ;5544
```

```
            ;5545  .PAGE "TEST 1CC RESERVED"
            ;5546 =0
U 1210, 0000,003D,0180,0800,0000,1093  ;5547 T1CC: NEWTST
U 1211, 0000,003C,0180,0800,0000,1212  ;5548  NOP
            ;5549
```

```
            ;5550  .PAGE "TEST 1CD RESERVED"
            ;5551 =0
U 1212, 0000,003D,0180,0800,0000,1093  ;5552 T1CD: NEWTST
U 1213, 0000,003C,0180,0800,0000,1214  ;5553  NOP
            ;5554
```

```
            ;5555  .PAGE "TEST 1CE RESERVED"
            ;5556 =0
U 1214, 0000,003D,0180,0800,0000,1093  ;5557 T1CE: NEWTST
U 1215, 0000,003C,0180,0800,0000,13DC  ;5558  NOP
            ;5559
```

```
            ;5560  .PAGE "DUMMY NEWTST"
U 13DC. 0000,003D,0180,0800,0000,1093  ;5561 DUM: NEWTST
            ;5562
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1464
ESKAR4C.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:13     Page  15
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1929: 1908 1917= 1918= 2762= 2763= 2764= 2767=
U 1008  2363= 2365= 2367= 2372= 2352= 2353= 2354= 2355=
U 1010  2386= 2388= 2389= 2393= 2409= 2410= 2411= 2419=
U 1018  2889= 2890= 2891= 2892= 2893= 2894= 2895= 2896=
U 1020  2421= 2423= 2425= 2426= 2469= 2470= 2471= 2473=
U 1028  2517= 2518= 2519= 2521= 3264= 3265= 3266= 3267=
U 1030  3313= 3314= 3321= 3323= 3447= 3448= 3449= 1909
U 1038  3457= 3458= 3459= 3460= 3552= 3553= 3561= 3578=
U 1040  3738= 3739= 3740= 1910 3748= 3749= 3750= 3751=
U 1048  4133= 4134= 4135= 1911 1919= 1920= 2028: 1913
U 1050  4144= 4145= 4146= 4147= 4453= 4454= 4455= 1914
U 1058  1972= 1974= 2074: 1921 1985= 1986= 2108: 1922
U 1060  4464= 4465= 4466= 1923 4468= 4470= 4473= 4475=
U 1068  4809= 4810= 4811= 1924 4820= 4821= 4822= 4823=
U 1070  4854= 4855= 4863= 4881= 4926= 4927= 4928= 1925
U 1078  4958= 4959= 4967= 4968= 5041= 5042= 5049= 5067=
U 1080  5110= 5111= 5112= 1926 5140= 5141= 5149= 5150=
U 1088  5218= 5219= 5226= 5244= 5289= 5290= 5291= 1927
U 1090  5319= 5320= 5321= 1961 5401= 5402= 5403= 1962
U 1098  5476= 5477= 5486= 5487= 2046= 2047= 2092= 2093=
U 10A0  2215= 2216= 2341= 2342= 2344= 2349= 2579= 2584=
U 10A8  2589= 2594= 2599= 2600= 2644= 2645= 2722= 2723=
U 10B0  2801= 2802= 2868= 2869= 2877= 2878= 2879= 2881=
U 10B8  2884= 2885= 2926= 2927= 2928= 2929= 2936= 2937=
U 10C0  2943= 2945= 2948= 2949= 2968= 2969= 2971= 2972=
U 10C8  3062= 3063= 3066= 3067= 3078= 3079= 3082= 3083=
U 10D0  3212= 3213= 3387= 3388= 3389= 3390= 3391= 3393=
U 10D8  3445= 3446= 3461= 3462= 3463= 3464= 3473= 3474=
U 10E0  3484= 3500= 3514= 3515= 3524= 3525= 3542= 3543=
U 10E8  3544= 3546= 3547= 3548= 3550= 3551= 3588= 3589=
U 10F0  3600= 3601= 3618= 3619= 3620= 3621= 3643= 3644=
U 10F8  3664= 3665= 3736= 3737= 3752= 3753= 3754= 3755=
U 1100  1894: 1895: 1896: 1897: 1898: 1899: 1900: 1901:
U 1108  1902: 1963 3769= 3770= 1903: 1904: 1905: 1906:
U 1110  3780= 3801= 3803= 3809= 3811= 3817= 3819= 3825=
U 1118  3827= 3828= 3837= 3838= 3839= 3840= 3841= 3842=
U 1120  3843= 3845= 3846= 3847= 3857= 3858= 3868= 3884=
U 1128  3891= 3892= 3901= 3902= 3919= 3920= 3921= 3923=
U 1130  3924= 3925= 3936= 3937= 3946= 3962= 3969= 3970=
U 1138  3979= 3980= 3997= 3998= 3999= 4000= 4001= 4002=
U 1140  4016= 4017= 4028= 4029= 4049= 4050= 4131= 4132=
U 1148  4148= 4149= 4150= 4151= 4163= 4164= 4174= 4192=
U 1150  4201= 4207= 4217= 4223= 4233= 4239= 4249= 4250=
U 1158  4261= 4262= 4263= 4264= 4265= 4266= 4267= 4268=
U 1160  4270= 4271= 4273= 4274= 4281= 4282= 4291= 4304=
U 1168  4312= 4313= 4322= 4323= 4338= 4339= 4340= 4341=
U 1170  4342= 4343= 4355= 4356= 4366= 4381= 4384= 4385=
U 1178  4451= 4452= 4486= 4502= 4511= 4512= 4528= 4529=
U 1180  4530= 4531= 4533= 4534= 4535= 4536= 4537= 4538=
U 1188  4548= 4549= 4568= 4569= 4578= 4595= 4607= 4608=
U 1190  4617= 4634= 4650= 4651= 4660= 4677= 4682= 4683=
U 1198  4687= 4688= 4704= 4705= 4711= 4712= 4807= 4808=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4838= 4839= 4845= 4846= 4847= 4848= 4849= 4850=
U 11A8  4852= 4853= 4889= 4890= 4899= 4900= 4917= 4918=
U 11B0  4919= 4920= 4921= 4923= 4924= 4925= 4939= 4940=
U 11B8  4986= 4987= 4991= 4992= 5001= 5002= 5024= 5025=
U 11C0  5031= 5032= 5033= 5034= 5035= 5036= 5039= 5040=
U 11C8  5083= 5084= 5101= 5102= 5103= 5104= 5105= 5106=
U 11D0  5108= 5109= 5122= 5139= 5168= 5169= 5182= 5183=
U 11D8  5205= 5206= 5207= 5208= 5209= 5210= 5211= 5212=
U 11E0  5216= 5217= 5260= 5261= 5278= 5279= 5280= 5281=
U 11E8  5282= 5283= 5287= 5288= 5301= 5318= 5331= 5332=
U 11F0  5350= 5351= 5365= 5366= 5388= 5389= 5390= 5391=
U 11F8  5392= 5393= 5394= 5395= 5399= 5400= 5413= 5431=
U 1200  5447= 5448= 5465= 5466= 5467= 5468= 5469= 5470=
U 1208  5474= 5475= 5505= 5506= 5520= 5521= 5541= 5542=
U 1210  5547= 5548= 5552= 5553= 5557= 5558= 1964 1965
U 1218  1966 1967 1968 1969 1970 1971 1975 1976
U 1220  1977 1978 1979 1980 1981 1982 1983 1984
U 1228  2003 2026 2027 2072 2073 2156 2157 2158
U 1230  2176 2178 2204 2205 2206 2207 2208 2209
U 1238  2210 2211 2213 2214 2350 2351 2376 2396
U 1240  2397 2398 2405 2545 2546 2547 2548 2549
U 1248  2550 2574 2622 2646 2647 2648 2649 2667
U 1250  2668 2669 2670 2688 2689 2690 2691 2717
U 1258  2751 2753 2754 2755 2757 2758 2759 2760
U 1260  2761 2768 2769 2770 2771 2794 2795 2796
U 1268  2797 2799 2800 2829 2830 2831 2870 2871
U 1270  2872 2873 2882 2883 2887 2888 2901 2902
U 1278  2904 2905 2906 2908 2913 2914 2915 2917
U 1280  2922 2924 2925 2933 2934 2938 2939 2940
U 1288  2941 2942 2950 2951 2952 2954 2955 2956
U 1290  2957 2958 2959 2964 2966 2967 2970 2995
U 1298  2996 2997 2998 3026 3027 3029 3030 3031
U 12A0  3058 3060 3061 3065 3069 3070 3071 3072
U 12A8  3074 3075 2134: 3076 3077 3080 3081 3106
U 12B0  3107 3108 3109 3110 3111 3158 3160 3161
U 12B8  3162 3164 3166 3167 3169 3171 3172 3173
U 12C0  3174 3177 3178 3179 3182 3207 3208 3210
U 12C8  3211 3258 3259 3260 3262 3263 3307 3308
U 12D0  3309 3310 3311 3325 3327 3382 3383 3385
U 12D8  3386 3395 3465 3466 3467 3468 3471 3472
U 12E0  3501 3502 3503 3504 3505 3506 3507 3509
U 12E8  3512 3513 3579 3580 3581 3583 3586 3587
U 12F0  3622 3623 3624 3625 3626 3627 3631 3632
U 12F8  3756 3757 3758 3760 3761 3763 3764 3765
U 1300  3767 3768 3802 3810 3818 3826 3848 3849
U 1308  3850 3851 3852 3853 3855 3856 3885 3886
U 1310  3889 3890 3926 3927 3928 3929 3930 3931
U 1318  3934 3935 3963 3964 3967 3968 4003 4004
U 1320  4005 4007 4008 4009 4010 4011 4012 4014
U 1328  4015 4152 4153 4154 4155 4156 4157 4158
U 1330  4159 4161 4162 4193 4194 4195 4196 4198
U 1338  4199 4208 4209 4210 4211 4213 4214 4215
U 1340  4224 4225 4226 4227 4229 4230 4231 4240
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4241 4242 4243 4245 4246 4247 4269 4272
U 1350  4275 4276 4277 4279 4280 4307 4308 4310
U 1358  4311 4344 4345 4346 4347 4348 4349 4350
U 1360  4351 4353 4354 4507 4513 4514 4518 4520
U 1368  4521 4522 4532 4539 4541 4542 4543 4545
U 1370  4546 4547 4551 4552 4553 4554 4555 4556
U 1378  4558 4560 4561 4562 4563 4564 4566 4567
U 1380  4596 4599 4600 4601 4602 4603 4605 4606
U 1388  4635 4638 4639 4640 4642 4644 4645 4646
U 1390  4648 4649 4678 4680 4681 4684 4685 4686
U 1398  4689 4692 4694 4696 4698 4699 4700 4702
U 13A0  4703 4824 4825 4826 4827 4828 4829 4830
U 13A8  4831 4840 4841 4842 4843 4851 4882 4883
U 13B0  4884 4885 4887 4888 4988 4989 4990 5026
U 13B8  5027 5028 5029 5038 5068 5069 5070 5071
U 13C0  5073 5074 5107 5170 5172 5173 5214 5245
U 13C8  5246 5247 5248 5250 5251 5285 5352 5354
U 13D0  5355 5397 5432 5433 5434 5435 5437 5438
U 13D8  5472 5507 5509 5510 5561
U 13E0 - 13EF Unused
U 13F0  2118: 2119: 2120: 2121: 2122: 2123: 2124: 2125:
U 13F8  2126: 2127: 2128: 2129: 2130: 2131: 2132: 2133:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR4C	1005
Used	1005
Remaining	

Total microwords used in memory U: 1005
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4C.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1910 Micro Monitor Interface Routines
; 1911   Newtest Routine
; 1973   ERROR1 WITH PARAMETER
; 1998   ERROR 2 WITH PARAMETER
; 2082   NOINTR ROUTINE
; 2260   Wait Loop Routine
; 2292   FIND MS780-E MEMORY
; 2434   Generate IO Address
; 2461   Set Starting Address
; 2495   CONFIGURE MS780-E/H
; 2569   ENABLE NEXT MS780-E
; 2621   RESET SUBTEST NUMBER
; 2644   Initialize the SBI
; 2672   Select Controller
; 2710   SELECT LOWER CONTROLLER
; 2759   SELECT UPPER CONTROLLER
; 2807   Set Trap Mask
; 2836   Disable Cache Routine
; 2857 TEST 1CF PRELIMINARY CACHE TEST
; 3117 TEST 1D0 QUAD WORD PLACEMENT IN CACHE ON BYTE READS
; 3385 TEST 1D1 PRELIMINARY CACHE-MEMORY DATA TEST
; 3662 TEST 1D2 CACHE UPDATE ON WRITE (BYTE) HITS
; 3937 TEST 1D3 RESERVED
; 3942 TEST 1D4 RESERVED
; 3947 DUMMY NEWTST

```
:1  .NOLIST
:1804  .LIST
:1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR4D
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
            ;1841  .PAGE
            ;1842
            ;1843  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100.  0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101.  0000.003C.0580.0800.0084.7001  ;1862 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102.  0000.003C.0980.0800.0084.7001  ;1863 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103.  0000.003C.0D80.0800.0084.7001  ;1864 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104.  0000.003C.1180.0800.0084.7001  ;1865 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105.  0000.003C.D580.0800.0084.7001  ;1866 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106.  0000.003C.D980.0800.0084.7001  ;1867 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107.  0000.003C.5D80.0800.0084.7001  ;1868 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108.  0000.003C.0180.0800.0084.7001  ;1869 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C.  0000.003C.8580.0800.0084.7001  ;1870 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D.  0000.003C.8980.0800.0084.7001  ;1871 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E.  0000.003C.6580.0800.0084.7001  ;1872 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F.  0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001.  0000.003C.81F0.2C00.0000.1007  ;1874 TRPH0: Q_ID[USTACK]
U 1007.  0C00.003C.01E0.0800.0000.1013  ;1875  Q_D,D_Q
U 1013.  0000.003C.8180.3C00.0000.1016  ;1876  ID[USTACK]_D
U 1016.  0C00.003C.01E0.0800.0000.101B  ;1877  Q_D,D_Q
U 101B.  0000.003C.01C0.0A78.0000.101E  ;1878  Q_R[0F]
U 101E.  0001.2024.0180.0800.0010.1033  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1033.  0000.013C.0180.0800.0000.1002  ;1880  Z?    ; BRANCH IF NO
U 1002.  0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1014  ;1897 TRPH1: RC[0D]_SC
U 1014. 0000.003D.D980.0800.0084.718B  ;1898 =0 SETRCF[.9]
U 1015. 0000.003C.49F0.2C00.0000.1037  ;1899  Q_ID[TBER0]
U 1037. 0001.203C.4DF0.2DB0.0000.103B  ;1900  RC[6]_Q,Q_ID[TBER1]
U 103B. 0001.203C.65F0.2DB8.0000.103F  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 103F. 0001.203C.6DF0.2DD8.0000.1043  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1043. 0001.203C.75F0.2DE0.0000.1047  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1047. 0001.203C.79F0.2DC8.0000.104B  ;1904  RC[9]_Q,Q_ID[PARITY]
U 104B. 0001.203C.69F0.2DC0.0000.104F  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 104F. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.1183  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908
```

```
                ;1909  .PAGE
                ;1910  .TOC " Micro Monitor Interface Routines"
                ;1911  .TOC "   Newtest Routine"
                ;1912 ;++
                ;1913 ; FUNCTIONAL DESCRIPTION:
                ;1914 ;
                ;1915 ; This routine Initializes the following registers:
                ;1916 ;  PSL    to 1F
                ;1917 ;  CES    to 0
                ;1918 ;  ACC.CS    to disable accellerator
                ;1919 ;  SIR    to 0
                ;1920 ;  CLK.CS    to stop interval timer
                ;1921 ;  TBER0    to disable the TB
                ;1922 ;  SBI.MAINT to     0
                ;1923 ;  SBI.ERR    to 0
                ;1924 ;  FAULT    to 0
                ;1925 ;  COMP    to 0
                ;1926 ;  RC[0E]    to  0
                ;1927 ;  R[0F]    to 0
                ;1928 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1929 ; A SCOPE call is then made to the Monitor.
                ;1930 ;
                ;1931 ; CALLING CONSTRAINT:
                ;1932 ;
                ;1933 ; =0
                ;1934 ;
                ;1935 ; SIDE EFFECTS:
                ;1936 ;
                ;1937 ; D Reg is destroyed
                ;1938 ; SC is destroyed
                ;1939 ;--
U 1053, 0000,003C,65F8,0800,0084,7057  ;1940 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1057, 0818,0038,8D80,0800,0000,105B  ;1941  d_k[.1f]  ; D=1F
U 105B, 0D00,003C,0180,0800,0000,105F  ;1942  d_dal.sc  ; D=001F0000
U 105F, 0000,003C,3D80,3C00,0000,1063  ;1943  id[psl]_d  ; psl = 001F0000
U 1063, 081B,0038,0580,0800,0000,1067  ;1944  d_k[.1]  ; Clear the CES register
U 1067, 0D00,003C,0180,0800,0000,106B  ;1945  d_dal.sc  ; D = 00010000
U 106B, 0F00,003C,3180,3C00,0000,106F  ;1946  id[ces]_d,d_0  ; ces = 00010000
U 106F, 0000,003C,5D80,3C00,0000,1073  ;1947  id[acc.cs]_d  ; acc.cs = 0
U 1073, 0000,003C,3980,3C00,0000,1077  ;1948  id[sir]_d  ; sir = 0
U 1077, 0000,003C,2980,3C00,0000,107B  ;1949  id[clk.cs]_d  ; clk.cs = 0
U 107B, 0000,003C,4980,3C00,0000,101C  ;1950  id[tber0]_d  ; tber0 = 0
U 101C, 0000,003D,0180,0800,0000,11A0  ;1951 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1952  intrpt.strobe,  ; Strobe interrupts
U 101D, 0818,0038,C180,0800,4000,107F  ;1953      d_k[.ffff]  ; Setup to clear SBI error register and
U 107F, 0801,0028,6580,3C00,0000,1083  ;1954  id[sbi.err]_d,d_not.d ; and fault register
U 1083, 0F00,003C,6D80,3C00,0000,1087  ;1955  id[fault]_d,d_0  ; ...
U 1087, 0000,003C,6D80,3C00,0000,108B  ;1956  id[fault]_d  ; fault = 0
U 108B, 0000,003C,7580,3C00,0000,1109  ;1957  id[maint]_d  ; MAINT = 0
U 1109, 0000,003C,6580,3C00,0000,1154  ;1958  id[sbi.err]_d  ; sbi error reg = 0
U 1154, 0000,003C,7180,3C00,0000,117E  ;1959  id[comp]_d  ; comp reg = 0
U 117E, 0000,003C,E580,3C00,0000,117F  ;1960  id[T9]_d  ; Clear code for subroutine START.TEST
U 117F, 0001,003C,0180,09F0,0000,1180  ;1961  rc[0e]_d  ;
U 1180, 0001,003C,0180,0AF8,0000,1181  ;1962  r[0f]_d  ; Clear expected trap mask
U 1181, 0001,003C,0180,09F8,0000,104C  ;1963  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 104C, 0000,003D,0180,0800,0000,1211  ;1964 =0 call[disable.cache] ; Disable the cache
U 104D, 0F03,003C,0180,09E8,0000,105E  ;1965  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
           ;1966
           ;1967
U 1182, 0818,0038,0980,0800,0000,105E  ;1968 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
```

```
            ;1969  .PAGE
U 1183, 0810,0038,0180,0978,0000,1184  ;1970 ERROR1: D_RC[0F]
U 1184, 0819,0034,4D80,0800,0000,104E  ;1971  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;1972 104E: D_D.OR.K[.4],J/CALLCON
            ;1973  .TOC "   ERROR1 WITH PARAMETER"
            ;1974 ;++
            ;1975 ; FUNCTIONAL DESCRIPTION:
            ;1976 ;
            ;1977 ; This subroutine takes  as parameter tne number of arguments
            ;1978 ; to be printed to the console in the case of an error logout.
            ;1979 ;
            ;1980 ; CALLING CONSTRAINT:
            ;1981 ;
            ;1982 ; =0
            ;1983 ;INPUTS:
            ;1984 ;
            ;1985 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;1986 ;--
            ;1987 =0
            ;1988 ERR1.ARG:
U 1058, 0000,003D,0180,0800,0000,118B  ;1989  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1059, 0000,003C,0180,0800,0000,1183  ;1990  J/ERROR1  ; Go to ERROR1
            ;1991 ;
            ;1992 ;
            ;1993
```

```
            ;1994  .PAGE
U 1185, 0810,0038,0180,0978,0000,1186  ;1995 ERROR2: D_RC[0F]
U 1186, 0819,0034,4D80,0800,0000,105A  ;1996  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;1997 105A: D_D.OR.K[.6],J/CALLCON
            ;1998  .TOC "    ERROR 2 WITH PARAMETER"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2003 ; except that in this case after setting the number of arguments
            ;2004 ; too the console, control is transferred to routine ERROR2.
            ;2005 ;
            ;2006 ; INPUTS:
            ;2007 ;
            ;2008 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2009 ;
            ;2010 ;--
            ;2011 =0
            ;2012 ERR2.ARG:
U 105C, 0000,003D,0180,0800,0000,118B  ;2013  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 105D, 0000,003C,0180,0800,0000,1185  ;2014  J/ERROR2  ; Go to ERROR2
            ;2015 ;
            ;2016 ;
            ;2017
            ;2018 105E:
            ;2019 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2020  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2021 ;+
            ;2022 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2023 ;-
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2024 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2025 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2026 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2027 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2028 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2029 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2030 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2031 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2032 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2033 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2034 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2035 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2036 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2037 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2038 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;2039 13FF: NOP
U 1155, 0000,003C,0180,0800,0000,12AA  ;2040 1155: NOP
U 12AA, 0000,003C,0180,0800,0000,1187  ;2041 12AA: NOP
            ;2042 ;+
            ;2043 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2044 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2045 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2046 ; TYPING IT.
            ;2047 ;-
U 1187, 0810,0038,4D80,0978,0000,1188  ;2048 SUBTST: D_RC[0F],KMX/.FF00
```

```
U 1188, 0019,4016,4D80,09F8,0000,0001  ;2049  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2050 ;*****************************************************************
            ;2051 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2052 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2053 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2054 ;-
            ;2055 SUBTEST1:
U 1189, 0810,0038,0180,0978,0000,118A  ;2056  D_RC[0F]
U 118A, 0019,0032,4D80,09F8,0000,0001  ;2057  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2058 ;*
            ;2059 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2060 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2061 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2062 ;-
U 118B, 0010,0038,01C0,0978,0000,118C  ;2063 SETRCF: Q_RC[0F]
U 118C, 0019,2034,4DC0,0800,0000,118D  ;2064  Q_Q.AND.K[.FF00]
U 118D, 0019,2032,1D80,09F8,0000,0001  ;2065  RC[0F]_Q.OR.SC,RETURN1
            ;2066 ;*****************************************************************
            ;2067 ;*
            ;2068 ;      THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2069 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2070 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2071 ;
            ;2072 ; D AND SC GET CLOBBERED
            ;2073 ;-
            ;2074 ;*****************************************************************
U 118E, 0F00,003C,2180,0800,0084,718F  ;2075 ENBFLT: SC_K[.14],D_0
U 118F, 0000,003C,0580,0800,0084,B190  ;2076  SC_SC-K[.1]
U 1190, 0801,001C,0180,0800,0000,1191  ;2077  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 1191, 0600,003C,6D80,3C00,0000,1192  ;2078  ID[FAULT]_D,D_D.RIGHT
U 1192, 0000,003E,6D80,3C00,0000,0001  ;2079  ID[FAULT]_D,RETURN1
            ;2080
```

```
                ;2081  .PAGE
                ;2082  .TOC "   NOINTR ROUTINE"
                ;2083 ;*****************************************************************
                ;2084 ;*
                ;2085 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
                ;2086 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
                ;2087 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
                ;2088 ; IF ANY INTERRUPTS OCCURRED:
                ;2089 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
                ;2090 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
                ;2091 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
                ;2092 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
                ;2093 ;  RETURN1 IS USED TO GET BACK
                ;2094 ; ELSE
                ;2095 ;  RETURN2 IS USED
                ;2096 ;
                ;2097 ; THE D AND Q REGISTERS GET CLOBBERED.
                ;2098 ;
                ;2099 ;-
                ;2100 ;*****************************************************************
                ;2101 =0
U 108C, 0000,003D,0180,0800,0000,1198  ;2102 NOINTR: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 108D, 0000,003C,35F0,2C00,0000,1193  ;2103  Q_ID[VECTOR]
U 1193, 0019,2034,8180,0800,0010,1194  ;2104  ALU_Q.AND.K[.3FF],CLK.UBCC
U 1194, 0000,013C,0180,0800,0000,108E  ;2105  Z?
U 108E, 0000,003C,0180,0800,0000,119C  ;2106 =0 J/ERRFLT    ; INTR. OCCURRED,SET FLAG,GET DATA
U 108F, 0000,003E,0180,0800,0000,0002  ;2107  RETURN2     ; NO INTR, RETURN2
                ;2108 =0
U 1090, 0000,003D,0180,0800,0000,1198  ;2109 WRTOUT: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 1091, 0000,003C,35F0,2C00,0000,1195  ;2110  Q_ID[VECTOR]
U 1195, 0819,2034,8180,0800,0000,1196  ;2111  ALU_Q.AND.K[.3FF],D_ALU
U 1196, 0019,0020,A580,0800,0010,1197  ;2112  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 1197, 0000,013C,0180,0800,0000,1092  ;2113  Z?
U 1092, 0000,003C,0180,0800,0000,119C  ;2114 =0 J/ERRFLT    ; WRONG VECTOR,SET FLAG,GET DATA
U 1093, 0000,003E,0180,0800,0000,0002  ;2115  RETURN2     ; OK, RETURN2
U 1198, 0F00,003C,0180,0800,0000,1199  ;2116 STROBE: D_0      ; SET LOWEST IPL ACTIVE
U 1199, 0000,003C,3D80,3C00,0000,119A  ;2117  ID[PSL]_D     ;
U 119A, 0000,003C,0180,0800,4000,119B  ;2118  IEK/ISTR     ; STROBE INTERRUPT
U 119B, 0000,003E,0180,0800,0000,0001  ;2119  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 119C, 0018,0038,0580,09A8,0000,119D  ;2120 ERRFLT: RC[5]_K[.1]    ; Set Flag
U 119D, 0001,203C,65F0,2DC0,0000,119E  ;2121  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 119E, 0001,203C,6DF0,2DB8,0000,119F  ;2122  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 119F, 0001,203E,0180,09B0,0000,0001  ;2123  RC[6]_Q,RETURN1    ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
                ;2124
                ;2125 SBI.UNJAM:
                ;2126 CLRSBI:
                ;2127 ;=================================================================
                ;2128 ;/////////////////////////////////////////////////////////////////
                ;2129 ;*
                ;2130 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
                ;2131 ; HOLD IS ASSERTED FOR 16 CYCLES.
                ;2132 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
                ;2133 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
                ;2134 ;-
U 11A0, 0818,0038,6580,8000,0010,1094  ;2135 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
```

```
                ;2136 =0
U 1094, 0819.0100.0580.8000.0010.1094   ;2137 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 1095, 0818.0038.6580.8800.0010.1096   ;2138  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
                ;2139 =0
U 1096, 0819.0100.0580.8800.0010.1096   ;2140 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 1097, 0818.0038.6580.8000.0010.1098   ;2141  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
                ;2142 =0
U 1098, 0819.0100.0580.8000.0010.1098   ;2143 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 1099, 0000.003E.0180.0800.0000.0001   ;2144  RETURN1   ; DONE.
                ;2145 INIT: ;=================================================================
                ;2146  ;
                ;2147  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
                ;2148  ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
                ;2149  ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
                ;2150  ;
                ;2151  ;=================================================================
                ;2152  ;
U 11A1, 0003.003C.0180.0AF8.0000.109A   ;2153  R[0F]_0    ;CLEAR U-TRAPS
U 109A, 0000.003D.0180.0800.0000.11A0   ;2154 =0 CALL[CLRSBI]   ; EXECUTE AN UNJAM SEQUENCE
U 109B, 0000.003C.0180.0800.0000.109C   ;2155  NOP
U 109C, 0000.003D.0180.0800.0000.11A4   ;2156 =0 CALL[SETPSL]   ;SET PSL AND CLEAR SOFTWARE
                ;2157       ;  INTRUPT REG
U 109D, 0000.003C.0180.0800.0000.109E   ;2158  NOP
U 109E, 0000.003D.0180.0800.0000.11AB   ;2159 =0 CALL[CLRFLT]   ;CLEAR CPU FAULT BIT
U 109F, 0000.003C.0180.0800.0000.10A0   ;2160  NOP
U 10A0, 0000.003D.0180.0800.0000.11AF   ;2161 =0 CALL[CLRECC]   ;CLEAR ANY PENDING ECC INTRUPTS
U 10A1, 0000.003C.0180.0800.0000.10A2   ;2162  NOP
U 10A2, 0000.003D.0180.0800.0000.11A7   ;2163 =0 CALL[CLRTIM]   ;CLEAR TIMEOUT BIT
U 10A3, 0000.003C.0180.0800.0000.10A4   ;2164  NOP
U 10A4, 0000.003D.0180.0800.0000.11B7   ;2165 =0 CALL[ENBECC]   ;ENABLE ECC INTRUPTS
U 10A5, 0000.003C.0180.0800.0000.10A6   ;2166  NOP
U 10A6, 0000.003D.0180.0800.0000.118E   ;2167 =0 CALL[ENBFLT]   ;ENBALE FAULT INTERRUPTS
U 10A7, 0818.0038.4580.0800.0000.10A8   ;2168  D_K[.8000]   ;ENABLE CACHE PARITY ERROR
U 10A8, 0000.003D.7980.3C00.0000.11B3   ;2169 =0 CALL[CACOFF],ID[PARITY]_D ;SHUT CACHE OFF
U 10A9, 0F00.003C.0180.0800.0000.11A2   ;2170  D_0
U 11A2, 0000.003C.4980.3C00.0000.11A3   ;2171  ID[TBER0]_D   ;SHUT TB OFF
U 11A3, 0000.003E.7180.3C00.0000.0001   ;2172  ID[COMP]_D,RETURN1  ;DISABLE SILO INTERRUPTS
            ;2173  ;
            ;2174  ;=================================================================
            ;2175  ;+
            ;2176  ;CHECK FOR INTERRUPTS PENDING
            ;2177  ;
            ;2178  ; INTERRUPT  VECTOR CHECKED
            ;2179  ; ------------------------------------
            ;2180  ; WRITE TIMEOUT  ^X60
            ;2181  ; SBI FAULT  ^X5C
            ;2182  ; CRD,RDS TAG  ^X54
            ;2183  ; SILO    ^X50
            ;2184  ;
            ;2185  ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2186  ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2187  ;
            ;2188  ; DATA: EXPECTED VECTOR
            ;2189  ;  VECTOR STROBED
            ;2190  ;-
```

```
            ;2191  ;******************************************************************************
            ;2192  ;
            ;2193 SETPSL: ;*************************************************************************
            ;2194  ;
            ;2195  ;DROP THE CPU IPR LEVEL TO
            ;2196  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2197  ; PENDING
            ;2198  ;
            ;2199  ;******************************************************************************
            ;2200  ;
U 11A4, 0F00,003C,0180,0800,0000,11A5  ;2201  D_0
U 11A5, 0000,003C,3980,3C00,0000,11A6  ;2202  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 11A6, 0000,003E,3D80,3C00,0000,0001  ;2203  RETURN1,ID[PSL]_D
            ;2204 CLRTIM: ;*************************************************************************
            ;2205  ;
            ;2206  ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2207  ; SBI ERROR REGISTER
            ;2208  ;
            ;2209  ;******************************************************************************
            ;2210  ;
U 11A7, 0818,0038,0580,0800,0000,11A8  ;2211  D_K[.1]
U 11A8, 0000,003C,85F8,0800,0084,71A9  ;2212  Q_0,SC_K[.C]
U 11A9, 0D00,003C,0180,0800,0000,11AA  ;2213  D_DAL.SC
U 11AA, 0000,003E,6580,3C00,0000,0001  ;2214  ID[SBI.ERR]_D,RETURN1
            ;2215 CLRFLT: ;*************************************************************************
            ;2216  ;
            ;2217  ;CLEAR THE CP FAULT BIT IN THE
            ;2218  ; FAULT REGISTER
            ;2219  ;
            ;2220  ;******************************************************************************
            ;2221  ;
U 11AB, 0818,0038,0180,0800,0000,11AC  ;2222  D_K[.8]
U 11AC, 0000,003C,65F8,0800,0084,71AD  ;2223  Q_0,SC_K[.10]
U 11AD, 0D00,003C,0180,0800,0000,11AE  ;2224  D_DAL.SC
U 11AE, 0000,003E,6D80,3C00,0000,0001  ;2225  ID[FAULT]_D,RETURN1
            ;2226 CLRECC: ;*************************************************************************
            ;2227  ;
            ;2228  ;CLEAR CRD,RDS BIT IN THE
            ;2229  ;SBI ERROR REGISTER
            ;2230  ;
            ;2231  ;******************************************************************************
            ;2232  ;
U 11AF, 0818,0038,0D80,0800,0000,11B0  ;2233  D_K[.3]
U 11B0, 0000,003C,89F8,0800,0084,71B1  ;2234  Q_0,SC_K[.D]
U 11B1, 0D00,003C,0180,0800,0000,11B2  ;2235  D_DAL.SC
U 11B2, 0000,003E,6580,3C00,0000,0001  ;2236  ID[SBI.ERR]_D,RETURN1
            ;2237 CACOFF: ;*************************************************************************
            ;2238  ;
            ;2239  ;FORCES MISSES IN THE CACHE
            ;2240  ;   ^X18000 TO THE MAINT REGISTER
            ;2241  ;
            ;2242  ;******************************************************************************
            ;2243  ;
U 11B3, 0818,0038,0D80,0800,0000,11B4  ;2244  D_K[.3]
U 11B4, 0000,003C,61F8,0800,0084,71B5  ;2245  Q_0,SC_K[.F]   ;15 SHIFTS TO THE LEFT
```

```
U 11B5. 0D00.003C.0180.0800.0000.11B6  ;2246  D_DAL.SC
U 11B6. 0000.003E.7580.3C00.0000.0001  ;2247  ID[MAINT]_D,RETURN1
        ;2248 ENBECC: ;=================================================================
        ;2249  ;
        ;2250  ;ENABLE CRD,RDS INTERRUPTS
        ;2251  ;
        ;2252  ;=================================================================
        ;2253  ;
U 11B7. 0818.0038.0580.0800.0000.11B8  ;2254  D_K[.1]
U 11B8. 0000.003C.61F8.0800.0084.71B9  ;2255  Q_0,SC_K[.F]
U 11B9. 0D00.003C.0180.0800.0000.11BA  ;2256  D_DAL.SC
U 11BA. 0000.003E.6580.3C00.0000.0001  ;2257  ID[SBI.ERR]_D,RETURN1
        ;2258
```

```
            ;2259  .PAGE
            ;2260  .TOC "   Wait Loop Routine"
            ;2261 ;**
            ;2262 ; FUNCTIONAL DESCRIPTION:
            ;2263 ;
            ;2264 ; This routine is used to wait the specified number of machine cycles.
            ;2265 ; This routine is typically called to wait for an SBI write to
            ;2266 ; I/O space to complete.
            ;2267 ;
            ;2268 ; CALLING CONSTRAINT:
            ;2269 ;
            ;2270 ; =0
            ;2271 ;
            ;2272 ; INPUTS:
            ;2273 ;
            ;2274 ; d - Contains number of cycles to wait
            ;2275 ; alu.z condition code must not be set
            ;2276 ;
            ;2277 ; SIDE EFFECTS:
            ;2278 ;
            ;2279 ; d is destroyed
            ;2280 ;--
            ;2281 WAIT.LOOP:
U 11BB, 0018,0038,6180,0800,0010,10AA  ;2282  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2283     ; ...is not set
            ;2284 =0
            ;2285 W.LOOP:
            ;2286  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10AA, 0819,0100,0580,0800,0010,10AA  ;2287       j/W.LOOP
U 10AB, 0000,003E,0180,0800,0000,0001  ;2288  return1   ; exit
            ;2289
            ;2290
```

```
;2291  .PAGE
;2292  .TOC "   FIND MS780-E MEMORY"
;2293 ;++
;2294 ; FUNCTIONAL DESCRIPTION:
;2295 ;
;2296 ;          The diagnostic is designed to handle up to 4 (four)
;2297 ;     MS780-E/H's on the 780 SBI. This routine looks at all TR levels
;2298 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
;2299 ;     3D will hold the I/O base address of the MS780-E subsystems found
;2300 ;     on the SBI, and ID Register 3E will hold the Total number of
;2301 ;     MS780-E/H's found minus one. Also, it is to be noted that the
;2302 ;     first MS780-E found will have its starting address set to 0
;2303 ;     (zero).
;2304 ;     All the other MS780-E/H's found will have their starting address
;2305 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
;2306 ;     Reg 3E to decide if there are any more MS780-E and will take
;2307 ;     apropriate action.  Register RC[0E] is used as a pointer to the
;2308 ;     current MS780-E unit under test.
;2309 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
;2310 ;     set to maximum.
;2311 ;
;2312 ; CALLING CONSTRAINT:
;2313 ;
;2314 ; =00
;2315 ;
;2316 ; OUTPUTS:
;2317 ;
;2318 ; rc[0e] - Contains address of Configuration Register
;2319 ; r[0]  - Contains TR level
;2320 ;
;2321 ; SIDE EFFECTS:
;2322 ;
;2323 ; D register is destroyed.
;2324 ;--
;2325 =0
;2326 FIND.MS780E:
U 10AC. 0000,003D,0180,0800,0000,10CA  ;2327  call[sbi.init]  ; Make sure SBI is enabled
U 10AD. 0003,003C,0180,0988,0000,11BC  ;2328  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11BC. 0818,0038,1980,0800,0000,11BD  ;2329  d_k[zero]  ; Clear MS780-E/H counter
U 11BD. 0000,003C,F980,3C00,0000,11BE  ;2330  id[3e]_d  ; ...
U 11BE. 0018,0038,0580,0A80,0000,11BF  ;2331  r[0]_k[.1]  ; Init TR counter
U 11BF. 0F03,003C,0180,09F0,0000,10AE  ;2332  d_0,rc[0E]_0  ; Clear "Save" location for
;2333      ; ...CNFG A Reg
;2334 =0
;2335 FIND.MEM.LOOP:
U 10AE. 0800,003D,0180,0A00,0000,11E5  ;2336  d_r[0],call[generate.io]; Generate address of TR level
U 10AF. 0001,003C,0180,09F0,0200,10B0  ;2337  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10B0. 0000,003D,8980,0800,0084,720E  ;2338 =0 set.trap.mask[.d] ; Enable timeout micro traps
;2339  d[long]_cache.p, ; Read the specified tr level
U 10B1. 0000,003C,0180,C970,0000,11C0  ;2340   ic_rc[0e]  ; ...
U 11C0. 0000,003C,0180,0A78,0010,11C1  ;2341  alu_r[0f],clk.ubcc ; Check for no response
U 11C1. 0001,013C,0180,0880,0082,10B2  ;2342  z?,sc_d,le_ra[0] ; Branch if no adapter here
U 10B2. 0000,003C,0180,0800,0000,11C2  ;2343 =0 j/check.adpt.code ; Check for a memory
U 10B3. 0000,003C,0180,0800,0000,11E1  ;2344  j/find.mem.lp1  ; Try next tr
;2345 CHECK.ADPT.CODE:
```

```
U 11C2, 0000,003C,1D80,0800,1404,71C3  ;2346  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 11C3, 0000,163C,0180,0800,0000,1008  ;2347  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,71C4  ;2348 =1000 j/ms780.e,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,71C5  ;2349  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11E1  ;2350  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11E1  ;2351  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,71CA  ;2352  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11E1  ;2353  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,11CE  ;2354  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,11CE  ;2355  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2356   ;
          ;2357   ; Found an MS780-A or -C. Set the starting address
          ;2358   ; to maximum.
          ;2359   ;
U 11C4, 0818,0038,5D80,0800,0000,11C6  ;2360 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11C5, 0818,0038,0580,0800,0000,11C6  ;2361 MS780.C:d_k[.1]  ; Setup bits<16:14> for -C controller
          ;2362 MS780.COMMON:
U 11C6, 0819,0030,7180,0800,0000,11C7  ;2363  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11C7, 0000,003C,01F8,0803,0080,D1C8  ;2364  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11C8, 0D00,003C,0180,0800,0000,11C9  ;2365  d_dal.sc  ; Put data in bits<29:14>
          ;2366  cache.p_d[long], ; Set the starting address to maximum
U 11C9, 0000,003C,0180,A800,0000,11E1  ;2367       j/find.mem.lp1 ; and continue TR scan
          ;2368   ;
          ;2369   ; Found an MA780. Set the starting address to maximum
          ;2370   ;
          ;2371 MA780.MAX:
U 11CA, 0818,0038,39F8,0803,0000,11CB  ;2372  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11CB, 0D00,003C,0180,0803,0000,11CC  ;2373  d_dal.sc,va_va+4 ; Put in proper position
U 11CC, 0000,003C,0180,0803,0000,11CD  ;2374  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;2375  cache.p_d[long], ; Set starting address to maximum
U 11CD, 0000,003C,0180,A800,0000,11E1  ;2376       j/find.mem.lp1
          ;2377 ;
          ;2378 ; Found an MS780E
          ;2379 ;
          ;2380 FOUND.MS780E:
U 11CE, 0000,003C,F9F0,2C00,0000,11CF  ;2381  q_id[3e]  ; Set up to see how many MS780-E's
          ;2382  alu_q.xor.k[.3], ; Are there Maximum units?
U 11CF, 0019,2020,0D80,0800,0010,11D0  ;2383   clk.ubcc ; Check condition codes
U 11D0, 0000,013C,0180,0800,0000,10B4  ;2384  z?   ; Are we at maximum units for system
U 10B4, 0000,003C,0180,0800,0000,11D1  ;2385 =0 J/ms780e.tst  ; No go find how many units ther are
U 10B5, 0818,0038,C180,0800,0000,10B6  ;2386  d_k[.ffff]  ; Yes set address to maximum
U 10B6, 0000,003D,0180,0800,0000,11E9  ;2387 =0 call[set.start.addr] ; .....
U 10B7, 0000,003C,0180,0800,0000,11E1  ;2388  j/find.mem.lp1  ; Go check to see if we are done
          ;2389 ;
          ;2390 MS780E.TST:
          ;2391  alu_rc[01],  ; Check "Found MS780E Flag"
U 11D1, 0010,0038,0180,0908,0010,11D2  ;2392   clk.ubcc ; Check condition codes
U 11D2, 0810,0138,0180,0970,0000,10B8  ;2393  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;2394 =0 j/not.first.ms780e, ; No
U 10B8, 0818,0038,C180,0800,0000,10BC  ;2395   d_k[.ffff] ; Set starting address
U 10B9, 0001,003C,0180,0988,0000,11D3  ;2396  rc[01]_d  ; Yes put base address in rc[01]
U 11D3, 0818,0038,7980,0800,0000,11D4  ;2397  d_k[.30]  ; Set up SC to point to first unit
U 11D4, 0019,0014,F580,0800,0082,11D5  ;2398  alu_d+k[.a],sc_alu ; ...
U 11D5, 0810,0038,0180,0908,0000,11D6  ;2399  d_rc[01]  ; Put base address of unit in D
U 11D6, 0000,003C,0180,3400,0000,11D7  ;2400  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11D7, 0F00,003C,0180,0800,0000,10BA  ;2401  d_0   ; Prepare to set starting address to Zero
U 10BA, 0000,003D,0180,0800,0000,11E9  ;2402 =0 call[set.start.addr] ; Go do it
           ;2403  j/find.mem.lp1,  ; Check to see if we are done
U 10BB, 0003,003C,0180,09E8,0000,11E1  ;2404   rc[0d]_0 ; ....
           ;2405 =0
           ;2406 NOT.FIRST.MS780E:
U 10BC, 0000,003D,0180,0800,0000,11E9  ;2407  call[set.start.addr] ; Go set starting address to maximum
U 10BD, 0000,003C,F9F0,2C00,0000,11D8  ;2408  q_id[3e]  ; Get the number of MS780-Es found
U 11D8, 0819,2014,0580,0800,0000,11D9  ;2409  d_q+k[.1]  ; Increment this counter
U 11D9, 0000,003C,F980,3C00,0000,11DA  ;2410  id[3e]_d  ; Save the counter
U 11DA, 0018,0038,11C0,0800,0000,11DB  ;2411  q_k[.4]   ; Prepare to form pointer to the ID registers
           ;2412      ; ...were the I/O base addresses will be stored
U 11DB, 081D,2000,7980,0800,0000,11DC  ;2413  d_q-d,k[.30]  ; ... adjust pointer
U 11DC, 0819,0014,7980,0800,0000,11DD  ;2414  d_d+k[.30]  ; Point the SC to the ID register
U 11DD, 0000,003C,F580,0800,0000,11DE  ;2415  k[.a]   ; ...to receive I/O base address
U 11DE, 0019,0014,F580,0800,0082,11DF  ;2416  alu_d+k[.a],sc_alu ; ...
U 11DF, 0810,0038,0180,0970,0000,11E0  ;2417  d_rc[0e]  ; Get base address to d
U 11E0, 0000,003C,0180,3400,0000,11E1  ;2418  id(sc)_d  ; Move base address to ID pointed to by SC
           ;2419   ;
           ;2420   ; Try next tr
           ;2421   ;
           ;2422 FIND.MEM.LP1:
U 11E1, 0818,0014,0580,0A80,0000,11E2  ;2423  r[0]_ia+k[.1],d_alu ; Increment TR level
           ;2424  alu_d.and.k[.f],
U 11E2, 0019,0034,6180,0800,0010,11E3  ;2425      clk.ubcc  ; Check if all done
U 11E3, 0000,013C,0180,0800,0000,10BE  ;2426  z?   ; Branch if yes
U 10BE, 0000,003C,0180,0800,0000,10AE  ;2427 =0 j/find.mem.loop  ; Try next TR
U 10BF, 0810,0038,0180,0908,0010,11E4  ;2428  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11E4, 0000,013C,0180,0800,0000,10C0  ;2429  z?   ; Check condition codes
U 10C0, 0001,003E,0180,09F0,0000,0002  ;2430 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10C1, 0000,003E,0180,0800,0000,0001  ;2431  return1   ; No MS780E found
           ;2432
```

```
            ;2433  .PAGE
            ;2434  .TOC "   Generate IO Address"
            ;2435 ;++
            ;2436 ; FUNCTIONAL DESCRIPTION:
            ;2437 ;
            ;2438 ; This routine is used to generate an SBI Configuration Register
            ;2439 ; address from a TR level.
            ;2440 ;
            ;2441 ; CALLING CONSTRAINT:
            ;2442 ;
            ;2443 ; =0
            ;2444 ;
            ;2445 ; INPUTS:
            ;2446 ;
            ;2447 ; D - Contains TR level
            ;2448 ;
            ;2449 ; OUTPUTS:
            ;2450 ;
            ;2451 ; D - Contains SBI IO address
            ;2452 ;--
            ;2453 GENERATE.IO:
U 11E5, 0000,003C,89F8,0800,0084,71E6  ;2454  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11E6, 0D00,003C,8D80,0800,0084,71E7  ;2455  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11E7, 0000,003C,0980,0800,0084,B1E8  ;2456  sc_sc-k[.2]  ; insert bit 29
U 11E8, 0801,001E,0180,0800,0000,0001  ;2457  d_d.ornot.mask,return1 ; and return
            ;2458
            ;2459
```

```
            ;2460   .PAGE
            ;2461   .TOC "   Set Starting Address"
            ;2462 ;**
            ;2463 ; FUNCTIONAL DESCRIPTION:
            ;2464 ;
            ;2465 ; This routine is used to set the starting address of the
            ;2466 ; memory to the specified value.
            ;2467 ;
            ;2468 ; CALLING CONSTRAINT:
            ;2469 ;
            ;2470 ; =0
            ;2471 ;
            ;2472 ; INPUTS:
            ;2473 ;
            ;2474 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2475 ;     (B Register Image divided by 2**16)
            ;2476 ; rc[0e] - Contains Address of Register A
            ;2477 ;
            ;2478 ; OUTPUTS:
            ;2479 ;
            ;2480 ; d - Contains aligned starting address (B Register Image)
            ;2481 ;
            ;2482 ; SIDE EFFECTS:
            ;2483 ;
            ;2484 ; d,q,va, and sc are destroyed
            ;2485 ;--
            ;2486 SET.START.ADDR:
U 11E9, 0010,0038,6580,0970,0284,71EA  ;2487  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11EA, 0000,003C,01F8,0803,0000,11EB  ;2488  va_va+4,q_0  ; Set address to Register B
            ;2489  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11EB, 0D00,003C,0980,0800,0084,B1EC  ;2490       sc_sc-k[.2]  ; Setup to insert bit 14
U 11EC, 0801,001C,0180,0800,0000,11ED  ;2491  d_d.ornot.mask  ; Insert Write Enable bit
U 11ED, 0000,003E,0180,A800,0000,0001  ;2492  cache.p_d[long],return1 ; Set the starting address and return
            ;2493
```

```
            ;2494  .PAGE
            ;2495  .TOC "   CONFIGURE MS780-E/H"
            ;2496 ;*************************************************************
            ;2497 ;**
            ;2498 ;
            ;2499 ; Functional Description:
            ;2500 ; ----------------------
            ;2501 ;
            ;2502 ;      This  subroutine  will  be  used  by  tests  that   do   not
            ;2503 ; specifically  check  the  memory, but  make use of it to execute.
            ;2504 ; Its purpose is to find a MS780-E and  configure  it  properly  so
            ;2505 ; that  if  a  Read/Write operation will take place no false errors
            ;2506 ; will be logged due to misconfigured memory.
            ;2507 ;
            ;2508 ; Calling Constraint: =00
            ;2509 ; ------------------
            ;2510 ;
            ;2511 ;
            ;2512 ; Inputs:
            ;2513 ; ------
            ;2514 ;
            ;2515 ; RC[0E] - I/O BASE OF MS780-E/H
            ;2516 ;
            ;2517 ;
            ;2518 ; Outputs:
            ;2519 ; -------
            ;2520 ;
            ;2521 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
            ;2522 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
            ;2523 ;      OR IF NO MS780-E/Hs WERE FOUND
            ;2524 ;      ON THE SBI
            ;2525 ;
            ;2526 ; Side Effects:
            ;2527 ; ------------
            ;2528 ;
            ;2529 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
            ;2530 ;
            ;2531 ;
            ;2532 ;
            ;2533 ;--
            ;2534 ;*************************************************************
            ;2535 CONFIG.MS780E:
U 11EE, 0818,0038,0D80,0800,0000,11EF  ;2536  D_K[.3]    ; Set up flag for the Fail Chain
U 11EF, 0001,003C,0180,09E8,0000,1004  ;2537  RC[0D]_D    ; ...
U 1004, 0000,003D,0180,0800,0000,10AC  ;2538 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1005, 0000,003D,0980,0800,0084,7058  ;2539  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 1006, 0000,003C,0180,0800,0000,1010  ;2540  NOP     ; OK. Found at least one MS780-E/H
            ;2541 =
            ;2542 CONFIG.RETRY:     ; Entry point for the case in which the
            ;2543       ; ...first MS780-E/H could not be
            ;2544       ; ...properly configured
U 1010, 0000,003D,0180,0800,0000,1024  ;2545 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1011, 0000,003C,0180,0800,0000,1018  ;2546  J/CNFG.LMIS    ; Lower controller misconfigured
U 1012, 0000,003E,0180,0800,0000,0001  ;2547  RETURN1     ; OK. Lower Controller is configured
            ;2548 =
```

```
            ;2549 =00
            ;2550 CNFG.LMIS:
U 1018. 0000,003D,0180,0800,0000,1028  ;2551  CALL[SELECT.UPPER]  ; Select the upper controller
U 1019. 0000,003C,0180,0800,0000,1020  ;2552  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 101A. 0000,003E,0180,0800,0000,0001  ;2553  RETURN1    ; OK. Upper Controller is configured
            ;2554 =
            ;2555 =00
            ;2556 CNFG.UMIS:
U 1020. 0000,003D,0180,0800,0000,11F1  ;2557  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2558       ; ...MS780-E/H found so go and see
            ;2559       ; ...if there are any more
U 1021. 0000,003C,0180,0800,0000,11F0  ;2560  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2561       ; ...try to configure it
U 1022. 0818,0038,0D80,0800,0000,1023  ;2562  D_K[.3]    ; Set up Flag for the Fail Chain
U 1023. 0001,003C,0180,09E8,0000,11F0  ;2563  RC[0D]_D   ; ...
U 11F0. 0000,003D,0980,0800,0084,7058  ;2564  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2565       ; ...abort the test
            ;2566 =
            ;2567
```

```
          ;2568  .PAGE
          ;2569  .TOC "   ENABLE NEXT MS780-E"
          ;2570 ;**
          ;2571 ; FUNCTIONAL DESCRIPTION:
          ;2572 ;
          ;2573 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
          ;2574 ;    They will be tested  separately, according to the TR level at which
          ;2575 ;    they are located,  starting with the one  at the lowest level first.
          ;2576 ;          When  a test has finished  with the first unit, it will have to call
          ;2577 ;    this specific  routine  to see  if any other MS780-E unit is present
          ;2578 ;    on the SBI. If more units are  present, the first one  will be  dis-
          ;2579 ;    abled  by  setting  its  starting  address to maximum, the next unit
          ;2580 ;    will be  enabled  by  setting its starting address to 0 and the test
          ;2581 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
          ;2582 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
          ;2583 ;    ID registers 3A through 3D and  a count of the Total number of units
          ;2584 ;    found - 1 in register 3E. In this subroutine the SC register is used
          ;2585 ;    as  a  pointer  to  ID  registers 3A through 3D.
          ;2586 ;
          ;2587 ; CALLING CONSTRAINT:
          ;2588 ;
          ;2589 ;  =00
          ;2590 ;
          ;2591 ;--
          ;2592 ENABLE.NEXT.MS780E:
U 11F1, 0000,003C,F9F0,2C00,0000,11F2  ;2593  Q_ID[3E]  ; Get total number of MS780E to Q
          ;2594  ALU_Q,   ; Check to see if it is zero
U 11F2, 0001,203C,0180,0800,0010,11F3  ;2595     CLK.UBCC  ; Check Condition Codes
U 11F3, 0000,013C,0180,0800,0000,10C2  ;2596  Z?   ; ...for zero
U 10C2, 0000,003C,0180,0800,0000,11F4  ;2597 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10C3, 0000,003E,0180,0800,0000,0002  ;2598  RETURN2   ; Zero No more units to test
          ;2599 ENABLE.NEXT:
U 11F4, 0818,0038,C180,0800,0000,10C4  ;2600  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10C4, 0000,003D,0180,0800,0000,11E9  ;2601 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10C5, 0818,0038,7980,0800,0000,11F5  ;2602  D_K[.30]  ; Prepare to point to the ID register holding
          ;2603     ; ...the I/O Base address of the next MS780-E
U 11F5, 0819,0014,1180,0800,0000,11F6  ;2604  D_D+K[.4]  ; ...
U 11F6, 0000,003C,F580,0800,0000,11F7  ;2605  K[.A]   ; ...
U 11F7, 0819,0014,F580,0800,0000,11F8  ;2606  D_D+K[.A]  ; ...
U 11F8, 0000,003C,F9F0,2C00,0000,11F9  ;2607  Q_ID[3E]  ; Get MS780-E Counter
          ;2608  D_D-Q,   ; D now holds the pointer to the ID register
U 11F9, 081D,0000,0180,0800,0082,11FA  ;2609    SC_ALU  ; ...
U 11FA, 0000,003C,01F0,2400,0000,11FB  ;2610  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11FB, 0001,203C,0180,09F0,0000,11FC  ;2611  RC[0E]_Q  ; Save it also in RC[0E]
U 11FC, 0F03,003C,0180,09E8,0000,10C6  ;2612  RC[0D]_0,D_0  ; Set starting address to zero
U 10C6, 0000,003D,0180,0800,0000,11E9  ;2613 =0 CALL[SET.START.ADDR] ; ....
U 10C7, 0F00,003C,F9F0,2C00,0000,11FD  ;2614  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11FD, 081D,2008,0180,0800,0000,11FE  ;2615  D_Q-D-1   ; Subtract 1 from total
U 11FE, 0000,003C,F980,3C00,0000,10C8  ;2616  ID[3E]_D  ; Adjust the pointer
U 10C8, 0000,003D,0180,0800,0000,11FF  ;2617 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10C9, 0000,003E,0180,0800,0000,0001  ;2618  RETURN1   ; Tell caller another unit was found
          ;2619
```

```
          ;2620  .PAGE
          ;2621  .TOC "   RESET SUBTEST NUMBER"
          ;2622 ;++
          ;2623 ; FUNCTIONAL DESCRIPTION:
          ;2624 ;
          ;2625 ; Since some  of the tests will have to be restarted when two
          ;2626 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
          ;2627 ; the subtest counter to zero ( only for thoes tests that have
          ;2628 ; more than one subtest). This subroutine may also be necessary
          ;2629 ; when a test is being loop on.
          ;2630 ;
          ;2631 ; CALLING CONSTRAINT:
          ;2632 ;
          ;2633 ; =0
          ;2634 ; SIDE EFFECTS:
          ;2635 ;
          ;2636 ; D is destroyed
          ;2637 ;
          ;2638 ;--
          ;2639 RESET.SUBTST:
          ;2640  RC[0F]_0,   ; Reset subtest number
U 11FF, 0003,003E,0180,09F8,0000,0001  ;2641   RETURN1   ; ...and return
          ;2642
```

```
            ;2643  .PAGE
            ;2644  .TOC "   Initialize the SBI"
            ;2645 ;++
            ;2646 ; FUNCTIONAL DESCRIPTION:
            ;2647 ;
            ;2648 ; This routine performs the following functions:
            ;2649 ;
            ;2650 ;  Executes an SBI Unjam
            ;2651 ;  Clears the SBI Fault Latch
            ;2652 ;  Clears the SBI Error Register
            ;2653 ;
            ;2654 ; CALLING CONSTRAINT:
            ;2655 ;
            ;2656 ; =0
            ;2657 ;
            ;2658 ; SIDE EFFECTS:
            ;2659 ;
            ;2660 ; D & Q Register destroyed
            ;2661 ;--
            ;2662 =0
            ;2663 SBI.INIT:
U 10CA, 0000,003D,0180,0800,0000,11A0  ;2664  call[sbi.unjam]  ; Unjam the SBI
U 10CB, 0818,0038,C180,0800,0000,1200  ;2665  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1200, 0801,0028,6580,3C00,0000,1201  ;2666  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1201, 0F00,003C,6D80,3C00,0000,1202  ;2667  id[fault]_d,d_0
U 1202, 0000,003C,6D80,3C00,0000,1203  ;2668  id[fault]_d
U 1203, 0000,003E,6580,3C00,0000,0001  ;2669  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2670
```

```
            ;2671  .PAGE
            ;2672  .TOC "   Select Controller"
            ;2673 ;**
            ;2674 ; FUNCTIONAL DESCRIPTION:
            ;2675 ;
            ;2676 ; This routine is used to put the memory system into the
            ;2677 ; specified configuration with respect to controller select.
            ;2678 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2679 ; a RETURN2 is made.
            ;2680 ;
            ;2681 ; CALLING CONSTRAINT:
            ;2682 ;
            ;2683 ; =00
            ;2684 ;
            ;2685 ; INPUTS:
            ;2686 ;
            ;2687 ; d - Contains value to write to bits<2:0> of Register A
            ;2688 ; rc[0e] - Address of Register A
            ;2689 ;
            ;2690 ; SIDE EFFECTS:
            ;2691 ;
            ;2692 ; sc,d,va are destroyed
            ;2693 ;--
            ;2694 SELECT.CNTRLR:
U 1204, 0010,0038,0180,0970,0284,7205  ;2695  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1205, 0801,001C,0180,0800,0000,1206  ;2696  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1206, 0000,003C,0180,A800,0000,1207  ;2697  cache.p_d[long]  ; Write the configuration Register
U 1207, 0818,0038,5D80,0800,0000,1208  ;2698  d_k[.7]   ; Get Misconfiguration bits
U 1208, 0000,003C,61F8,0800,0084,7209  ;2699  sc_k[.F],q_0  ; ...
U 1209, 0D00,003C,0180,0800,0000,120A  ;2700  d_dal.sc  ; ... D = x38000
U 120A, 0000,003C,01E0,0800,0000,120B  ;2701  q_d   ; Save mask
U 120B, 0000,003C,0180,C800,0000,120C  ;2702  d[long]_cache.p  ; Read CNFG A Reg and
            ;2703  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 120C, 001D,2034,0180,0800,0010,120D  ;2704   clk.ubcc ; ...
U 120D, 0000,013C,0180,0800,0000,10CC  ;2705  z?   ; Branch if no
U 10CC, 0000,003E,0180,0800,0000,0001  ;2706 =0 RETURN1   ; Bad configuration
U 10CD, 0000,003E,0180,0800,0000,0002  ;2707  RETURN2   ; Configuration ok
            ;2708
```

```
;2709  .PAGE
;2710  .TOC "    SELECT LOWER CONTROLLER"
;2711 ;*************************************************************
;2712 ;**
;2713 ;
;2714 ; Functional Description:
;2715 ; ----------------------
;2716 ;
;2717 ; This subroutine can be called to select the lower
;2718 ; Controller on a MS780-E/H.
;2719 ; If the Lower controller is misconfigured then a
;2720 ; RETURN1 will be made.  Otherwise a RETURN2
;2721 ;
;2722 ; Calling Constraint: =00
;2723 ; ------------------
;2724 ;
;2725 ;
;2726 ; Inputs:
;2727 ; ------
;2728 ;
;2729 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2730 ;
;2731 ; Outputs:
;2732 ; -------
;2733 ;
;2734 ; RC[0D] will have the address of CNFG Register C
;2735 ;        ( 2000x008 )
;2736 ;
;2737 ; Side Effects:
;2738 ; ------------
;2739 ;
;2740 ; Contents of the following registers will lost:
;2741 ;
;2742 ; D
;2743 ;
;2744 ; The Lower controller on the MS780-E/H will be configured
;2745 ; when the subroutine is exitted with a RETURN2
;2746 ;--
;2747 ;*************************************************************
;2748 =00
;2749 SELECT.LOWER:
;2750   D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,1204  ;2751    CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2752  RETURN1     ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2753  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2754  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2755    RETURN2    ; Let caller know that the controller
;2756       ; ...was configured properly
;2757
```

```
;2758  .PAGE
;2759  .TOC "   SELECT UPPER CONTROLLER"
;2760 ;*************************************************************
;2761 ;++
;2762 ;
;2763 ; Functional Description:
;2764 ; ----------------------
;2765 ;
;2766 ; This subroutine can be called to select the upper
;2767 ; Controller on a MS780-E/H.
;2768 ; If the Upper controller is misconfigured then a
;2769 ; RETURN1 will be made.  Otherwise a RETURN2
;2770 ;
;2771 ; Calling Constraint: =00
;2772 ; ------------------
;2773 ;
;2774 ;
;2775 ; Inputs:
;2776 ; ------
;2777 ;
;2778 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2779 ;
;2780 ; Outputs:
;2781 ; -------
;2782 ;
;2783 ; RC[0D] will have the address of CNFG Register D
;2784 ;        ( 2000x010 )
;2785 ;
;2786 ; Side Effects:
;2787 ; ------------
;2788 ;
;2789 ; Contents of the following registers will lost:
;2790 ;
;2791 ; D
;2792 ;
;2793 ; The Upper controller on the MS780-E/H will be configured
;2794 ; when the subroutine is exitted with a RETURN2
;2795 ;--
;2796 ;*************************************************************
;2797 =00
;2798 SELECT.UPPER:
;2799  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,1204  ;2800   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2801  RETURN1    ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2802  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2803  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2804   RETURN2    ; Let caller know that the controller
;2805       ; ...was configured properly
```

```
            ;2806  .PAGE
            ;2807  .TOC "   Set Trap Mask"
            ;2808 ;++
            ;2809 ; FUNCTIONAL DESCRIPTION:
            ;2810 ;
            ;2811 ; This routine is used to set a bit in the Micro Trap Mask
            ;2812 ; R[0F].
            ;2813 ;
            ;2814 ; CALLING CONSTRAINT:
            ;2815 ;
            ;2816 ; =0
            ;2817 ;
            ;2818 ; INPUTS:
            ;2819 ;
            ;2820 ; SC - Contains bit number to set.
            ;2821 ;
            ;2822 ; OUTPUTS:
            ;2823 ;
            ;2824 ; rc[0E] - Contains the current trap mask
            ;2825 ;
            ;2826 ; SIDE EFFECTS:
            ;2827 ;
            ;2828 ; d regsiter is destroyed
            ;2829 ;--
            ;2830 SET_TRAP_MASK:
U 120E, 0800,003C,0180,0A78,0000,120F  ;2831  d_r[0f]
U 120F, 0801,001C,0180,0AF8,0000,1210  ;2832  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1210, 0001,003E,0180,0AF0,0000,0001  ;2833  r[0E]_d,return1    ; Save mask and return
            ;2834
```

```
            ;2835  .PAGE
            ;2836  .TOC "   Disable Cache Routine"
            ;2837 ;++
            ;2838 ; FUNCTIONAL DESCRIPTION:
            ;2839 ;
            ;2840 ; This routine disables cache hits by setting bits <16:15>
            ;2841 ; in the maintenance register.
            ;2842 ;
            ;2843 ; CALLING SEQUENCE:
            ;2844 ;
            ;2845 ; =0
            ;2846 ;
            ;2847 ; SIDE EFFECTS:
            ;2848 ;
            ;2849 ; D & Q Registers destroyed
            ;2850 ;--
            ;2851 DISABLE.CACHE:
U 1211, 0818,0038,4580,0800,0000,1212  ;2852  d_k[.8000]  ; ... and seed constant
U 1212, 0500,003C,0180,0800,0000,1213  ;2853  d_d.left  ; Generate final constant
U 1213, 0819,0030,4580,0800,0000,1214  ;2854  d_d.or.k[.8000]  ; ... = 00018000
U 1214, 0000,003E,7580,3C00,0000,0001  ;2855  id[maint]_d,return1 ; Disable cache and return
            ;2856
```

```
;2857   .PAGE "TEST 1CF PRELIMINARY CACHE TEST "
;2858  ;================================================================================
;2859  ;**
;2860  ;
;2861  ;
;2862  ; "CHET00.MIC"
;2863  ;
;2864  ; PRELIMINARY CACHE TEST, SIMPLE HITS AND MISSES,
;2865  ;    AND INVALIDATES
;2866  ;
;2867  ; TEST DESCRIPTION
;2868  ;
;2869  ; JUST CHECKS FOR CACHE HITS WITH READS TO MEMORY.
;2870  ; CHECKS THAT THE DATA IS FROM THE CACHE ON HITS, AND MEMORY ON
;2871  ; MISSES. THE "MCT/INVALIDATE" FUNCTION WILL BE CHECKED ALSO.
;2872  ;
;2873  ; SEQUENCE OF EVENTS ARE :
;2874  ; ------------------------------------------------
;2875  ;  NOTE: ADDRESS REMAINS CONSTANT
;2876  ;
;2877  ; 1) WRITE INVALIDATE TO CACHE
;2878  ; 2) SHUT CACHE OFF, WRITE TEST DATA
;2879  ;  TO MAIN MEMORY
;2880  ; 3) TURN ON CACHE, READ MAIN MEMORY
;2881  ;  SEE: MISS
;2882  ;       PROPER DATA (-1)
;2883  ; 4) READ MAIN MEMORY (2ND TIME)
;2884  ;  SEE: HIT
;2885  ;       PROPER DATA (-1)
;2886  ; 5) SHUT SBI OFF, READ MAIN MEMORY
;2887  ;  SEE: HIT
;2888  ;       PROPER DATA (-1)
;2889  ; 6) SHUT SBI OFF, WRITE HIT WITH
;2890  ;  F0F0F0F0 (RA 4),I.E. WRITE UPDATE
;2891  ; 7) TURN SBI ON, READ, SEE WHERE DATA COMES FROM
;2892  ;  SEE: F0F0F0F0 (IT'S FROM CACHE)
;2893  ;       HIT ALSO
;2894  ; 8) WRITE INVALIDATE TO CACHE
;2895  ; 9) READ MAIN MEMORY
;2896  ;  SEE: MISS
;2897  ;       -1 FOR DATA READ(IT'S FROM MEMEORY)
;2898  ;
;2899  ; LOGIC DESCRIPTION
;2900  ;
;2901  ; IF HITS,MISSES ARE IMPROPER THE SBL(M8218) OR SBH(M8219)  BOARDS
;2902  ; IF A FEW DATA BITS ARE BAD THE THE MEMORY ARRAY CARDS MAY BE
;2903  ;  CORRUPTING THE DATA AS ECC IS INHIBITED
;2904  ; CACHE PRITY CHECKER ARE ACTIVE WILL HITS ARE PRESENT, SO THEY
;2905  ;  MAY CAUSE PROBLEMS
;2906  ; THE MD BUS DRIVERS ON THE SBI,SBL BOARDS ARE ACTIVE.
;2907  ;
;2908  ; ERROR LOGOUT
;2909  ;
;2910  ;
;2911  ;    DATA: EXPECTED DATA
```

```
:2912 ;   DATA READ ( EITHER FROM CACHE OR MEMORY,
:2913 ;    SEE LISTING), ALSO HIT/MISS
:2914 ;    OPERATION MAY BE IN ERROR .
:2915 ;   GROUP FLAG(0=>GROUP 0, 1=> GROUP 1
:2916 ;   UNDEFINED
:2917 ;   UNDEFINED
:2918 ;   UNDEFINED
:2919 ;   ID[VECTOR] IFF UNEXPECTED INTERRUPT
:2920 ;   ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
:2921 ;   ID[FAULT] IFF UNEXPECTED INTERRUPT
:2922 ;   INTERRUPT FLAG:
:2923 ;    0=>NO INTERRUPT OCCURRED
:2924 ;    1=>AN INTERRUPT OCCURRED
:2925 ;
:2926 ; SYNC POINT DESCRIPTION
:2927 ;
:2928 ;
:2929 ;--
:2930 ;=================================================================
:2931 ;  SCRATCH PAD REQUIREMENTS
:2932 ; RA,B ------------------------------------
:2933 ; 0 SBI CYCLE DISABLE BIT   ^X1000
:2934 ; 1
:2935 ; 2 FORCE CACHE MIS BITS    ^X18000
:2936 ; 3 CACHE HIT BITS          ^X600
:2937 ; 4 CACHE DATA FOR READ TEST ^XF0F0F0F0
:2938 ; 5 MEMORY TEST DATA  ^XFFFFFFFF
:2939 ; F U-TRAP FLAG
:2940 ;
:2941 ; RC ---------------------------------------
:2942 ; 0 CONFIG REG A ADDRESS  ^X20002000
:2943 ; 1 CONFIG REG B ADDRESS  ^X20002004
:2944 ; 2 CONFIG REG C ADDRESS  ^X20002008
:2945 ; 3 BASE ADDRESS OF MEMORY  ^X0
:2946 ; E
:2947 ; F # OF ARG TO BE PASSED TO CONSOLE
:2948 ;
:2949 ;
:2950 ;
:2951 ;
:2952 ;*****************************************************************
:2953 ;
:2954 ;
:2955 =0
U 10CE, 0000.003D,0180,0800,0000,1053  ;2956 C00000: NEWTST
U 10CF, 0003.003C,0180,0AF8,0000,10D0  ;2957  R[0F]_0     ;CLEAR U-TRAPS
U 10D0, 0000.003D,0180,0800,0000,11EE  ;2958 =0 CALL[CONFIG.MS780E]  ; Configure MS780E (if any)
U 10D1, 0F00.003C,8580,0800,0084,7215  ;2959  D_0,SC_K[.C]   ;RA 0
U 1215, 0801.001C,0180,0A80,0000,1216  ;2960  R[0]_ALU,D_D.ORNOT.MASK
U 1216, 081B.0038,0DF8,0800,0000,1217  ;2961  D_K[.3],Q_0   ;R 2
U 1217, 0000.003C,6180,0800,0084,7218  ;2962  SC_K[.F]
U 1218, 0D00.003C,0180,0800,0000,1219  ;2963  D_DAL.SC
U 1219, 0001.003C,0180,0A90,0000,121A  ;2964  R[2]_D
U 121A, 081B.0038,D180,0800,0000,121B  ;2965  D_K[.C0]   ;RA 3
U 121B, 0100.003C,0180,0800,0000,121C  ;2966  D_D.LEFT2
```

```
U 121C. 0500.003C.0180.0800.0000.121D  ;2967  D_D.LEFT
U 121D. 0001.003C.0180.0A98.0000.121E  ;2968  R[3]_D
U 121E. 0818.0038.CDF8.0800.0000.121F  ;2969  D_K[.F0],Q_0   ;RA 4
U 121F. 0000.003C.0180.0800.0084.7220  ;2970  SC_K[.8]
U 1220. 0D00.003C.0180.0800.0000.1221  ;2971  D_DAL.SC
U 1221. 0819.0030.CD80.0800.0000.1222  ;2972  D_D.OR.K[.F0]
U 1222. 0D00.003C.0180.0800.0000.1223  ;2973  D_DAL.SC
U 1223. 0819.0030.CD80.0800.0000.1224  ;2974  D_D.OR.K[.F0]
U 1224. 0D00.003C.0180.0800.0000.1225  ;2975  D_DAL.SC
U 1225. 0819.0030.CD80.0AA0.0000.1226  ;2976  R[4]_ALU,D_D.OR.K[.F0]
U 1226. 0F00.003C.0180.0800.0000.1227  ;2977  D_0    ;RA 5
U 1227. 0801.0028.0180.0AA8.0000.1228  ;2978  R[5]_ALU,D_NOT.D
U 1228. 0003.003C.0180.0998.0000.1229  ;2979  RC[3]_0    ;RC 3
U 1229. 0018.0038.F580.09F8.0000.122A  ;2980  RC[0F]_K[.A]   ; Set Argument count
           ;2981  ;
           ;2982  ;=============================================================
           ;2983  ;
U 122A. 0003.003C.0180.09E0.0000.10D2  ;2984  RC[0C]_0   ;SET GROUP FLAG
U 10D2. 0000.003D.0180.0800.0000.1182  ;2985 =0 ERLOOP
U 10D3. 0000.003C.0180.0800.0000.10D4  ;2986  NOP
           ;2987 =0
           ;2988 C00020:
U 10D4. 0000.003D.0180.0800.0000.11A1  ;2989  CALL[INIT]   ; Clear Interrupts
U 10D5. 0818.0038.4580.0800.0000.122B  ;2990  D_K[.8000]
U 122B. 0010.0038.0180.0960.0010.122C  ;2991  ALU_RC[0C],CLK.UBCC  ;SET GROUP UNDER TEST
U 122C. 0600.013C.0180.0800.0000.10D6  ;2992  Z?,D_D.RIGHT
U 10D6. 0600.003C.0180.0800.0000.10D7  ;2993 =0 D_D.RIGHT   ;GROUP 1 UNDER TEST
U 10D7. 0010.0038.7580.3D18.0200.122D  ;2994  ID[MAINT]_D,VA_RC[3]  ;INVALIDATE THE CACHE,CLEAR
U 122D. 0000.003C.0180.9000.0000.122E  ;2995  MCT/INVALIDATE
U 122E. 0000.003C.0180.0803.0000.122F  ;2996  VA_VA+4
U 122F. 0000.003C.0180.9000.0000.10D8  ;2997  MCT/INVALIDATE
U 10D8. 0000.003D.0180.0800.0000.11B3  ;2998 =0 CALL[CACOFF]   ;SHUT CACHE OFF FOR WRITE TO
U 10D9. 0800.003C.0180.0A28.0000.10DA  ;2999  D_R[5]    ;  TO MEMORY (^XFFFF FFFF)
U 10DA. 0001.003D.F580.09F0.0084.718B  ;3000 =0 RC[0E]_D,SETRCF[.A]  ;SET EXPECTED DATA
U 10DB. 0010.0038.0180.0918.0200.1230  ;3001  VA_RC[3]   ;SET THE ADDRESS
U 1230. 0000.003C.0180.A800.0000.10DC  ;3002  MCT/WRITE.P   ;WRITE TEST DATA
U 10DC. 0818.0039.D580.0800.0000.11BB  ;3003 =0 D_K[.6],CALL[WAIT.LOOP]
U 10DD. 0000.003C.0180.0800.0000.102C  ;3004  NOP
U 102C. 0000.003D.0180.0800.0000.108C  ;3005 =00 CALL[NOINTR]
U 102D. 0000.003D.0180.0800.0000.1183  ;3006  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 102E. 0818.0038.4580.0800.0000.102F  ;3007  D_K[.8000]   ;SET GROUP UNDER TEST
U 102F. 0010.0038.0180.0960.0010.1231  ;3008  ALU_RC[0C],CLK.UBCC
U 1231. 0600.013C.0180.0800.0000.10DE  ;3009  Z?,D_D.RIGHT
U 10DE. 0600.003C.0180.0800.0000.10DF  ;3010 =0 D_D.RIGHT
U 10DF. 0000.003C.7580.3C00.0000.1232  ;3011  ID[MAINT]_D
U 1232. 0000.003C.0180.C800.0000.1233  ;3012  MCT/READ.P   ;READ MAIN MEMORY
U 1233. 0001.003C.0180.09E8.0000.1030  ;3013  RC[0D]_D   ;SAVE DATA READ
U 1030. 0000.003D.0180.0800.0000.108C  ;3014 =00 CALL[NOINTR]
U 1031. 0000.003D.0180.0800.0000.1183  ;3015  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1032. 0000.003C.75F0.2C00.0000.1234  ;3016  Q_ID[MAINT]   ;CHECK FOR CACHE MISS
U 1234. 000D.2034.0180.0A18.0010.1235  ;3017 = ALU_Q[AND]R[03],CLK.UBCC
U 1235. 0000.013C.0180.0800.0000.10E0  ;3018  Z?
U 10E0. 0000.003D.0180.0800.0000.1183  ;3019 =0 ERROR1    ; CACHE HIT , TEST FAILED
U 10E1. 0810.0038.0180.0968.0000.1236  ;3020  D_RC[0D]   ;CHECK THE DATA READ
U 1236. 080D.0020.0180.0A28.0010.1237  ;3021  D_ALU,ALU_D[XOR]R[5],CLK.UBCC
```

```
U 1237, 0000.013C.0180.0800.0000.10E2  ;3022  Z?
U 10E2, 0000.003D.0180.0800.0000.1183  ;3023 =0 ERROR1     ; DATA READ WITH CACHE MISS BAD
          ;3024       ;============================
          ;3025       ;DATA: EXPECTED DATA
          ;3026       ; DATA READ
          ;3027       ;============================
U 10E3, 0000.003C.0180.C800.0000.1238  ;3028  MCT/READ.P    ;READ DATA FOR 2ND TIME
U 1238, 0001.003C.0180.09E8.0000.1034  ;3029  RC[0D]_D    ;CHECK FOR CACHE HIT,CHECK DATA
U 1034, 0000.003D.0180.0800.0000.108C  ;3030 =00 CALL[NOINTR]
U 1035, 0000.003D.0180.0800.0000.1183  ;3031  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1036, 0000.003C.75F0.2C00.0000.1239  ;3032  Q_ID[MAINT]    ;CHECK FOR CACHE HIT
U 1239, 000D.2034.0180.0A18.0010.123A  ;3033 = ALU_Q[AND]R[03],CLK.UBCC
U 123A, 0000.013C.0180.0800.0000.10E4  ;3034  Z?
U 10E4, 0000.003C.0180.0800.0000.123B  ;3035 =0 J/C00040    ; CACHE HIT, TEST PASSED
U 10E5, 0000.003D.0180.0800.0000.1183  ;3036  ERROR1     ;CACHE MISS ON 2ND READ
U 123B, 0810.0038.0180.0968.0000.123C  ;3037 C00040: D_RC[0D]    ;CHECK DATA READ DURING CACHE HIT
U 123C, 000D.0020.0180.0A28.0010.123D  ;3038  ALU_D[XOR]R[5],CLK.UBCC
U 123D, 0000.013C.0180.0800.0000.10E6  ;3039  Z?
U 10E6, 0000.003D.0180.0800.0000.1183  ;3040 =0 ERROR1     ;DATA READ ON HIT BAD
          ;3041       ;============================
          ;3042       ;DATA: EXPECTED DATA
          ;3043       ; DATA READ
          ;3044       ;============================
U 10E7, 0000.003C.75F0.2C00.0000.123E  ;3045  Q_ID[MAINT]    ;DISABLE SBI,READ,SEE HIT
U 123E, 080D.2030.0180.0A00.0000.123F  ;3046  D_ALU,ALU_Q[OR]R[0]
U 123F, 0000.003C.7580.3C00.0000.1240  ;3047  ID[MAINT]_D    ; AND CHECK DATA READ
U 1240, 0000.003C.0180.C800.0000.1241  ;3048  MCT/READ.P    ;READ
U 1241, 0001.003C.0180.09E8.0000.1242  ;3049  RC[0D]_D    ;SAVE DATA READ
U 1242, 0000.003C.75F0.2C00.0000.1243  ;3050  Q_ID[MAINT]    ;CHECK FOR HIT
U 1243, 000D.2034.0180.0A18.0010.1244  ;3051  ALU_Q[AND]R[3],CLK.UBCC  ;CHECK CACHE HIT BITS
U 1244, 0000.013C.0180.0800.0000.10E8  ;3052  Z?
U 10E8, 0000.003C.0180.0800.0000.1245  ;3053 =0 J/C00070    ; CACHE HIT, TEST PASSED
U 10E9, 0000.003D.0180.0800.0000.1183  ;3054  ERROR1     ;CACHE MISS WHEN SBI DISABLED
U 1245, 0810.0038.0180.0968.0000.1246  ;3055 C00070: D_RC[0D]    ;CHECK THE DATA READ DURING HIT
U 1246, 000D.0020.0180.0A28.0010.1247  ;3056  ALU_D[XOR]R[5],CLK.UBCC
U 1247, 0000.013C.0180.0800.0000.10EA  ;3057  Z?
U 10EA, 0000.003D.0180.0800.0000.1183  ;3058 =0 ERROR1     ; DATA BAD FROM CACHE HIT
          ;3059       ;WHEN HIT AND SBI DISABLED
          ;3060       ;============================
          ;3061       ;DATA: EXPECTED DATA
          ;3062       ; DATA READ
          ;3063       ;============================
          ;3064       ;SBI IS DISABLED, WRITE TO CACHE
U 10EB, 0800.003C.0180.0A20.0000.1248  ;3065  D_R[04]     ;WRITE CACHE DIFFERENT THEN THAT
          ;3066       ;    IS IN MEMORY
U 1248, 0000.003C.0180.A800.0000.1249  ;3067  MCT/WRITE.P    ;WRITE WITH CACHE HIT
U 1249, 0818.0038.4580.0800.0000.124A  ;3068  D_K[.8000]    ;REENABLE SBI AND FORCE GROUP
U 124A, 0010.0038.0180.0960.0010.124B  ;3069  ALU_RC[0C],CLK.UBCC
U 124B, 0600.013C.0180.0800.0000.10EC  ;3070  Z?,D_D.RIGHT    ;GROUP 0 UNDER TEST
U 10EC, 0600.003C.0180.0800.0000.10ED  ;3071 =0 D_D.RIGHT    ;GROUP 1 UNDER TEST
U 10ED, 0000.003C.7580.3C00.0000.124C  ;3072  ID[MAINT]_D    ;SBI ON,CACHE ON ,READ
U 124C, 0000.003C.0180.C800.0000.124D  ;3073  MCT/READ.P    ;READ
U 124D, 0001.003C.0180.09E8.0000.124E  ;3074  RC[0D]_D    ;SAVE DATA READ
U 124E, 0000.003C.75F0.2C00.0000.124F  ;3075  Q_ID[MAINT]    ;SEE CACHE HIT
U 124F, 000D.2034.0180.0A18.0010.1250  ;3076  ALU_Q[AND]R[03],CLK.UBCC
```

```
U 1250, 0000.013C.0180.0800.0000.10EE  ;3077  Z?
U 10EE, 0000.003C.0180.0800.0000.1251  ;3078 =0 J/C00100   ; CACHE HIT ,TEST PASSED
U 10EF, 0000.003D.0180.0800.0000.1183  ;3079  ERROR1    ;CACHE MISS ,TEST FAILED
U 1251, 0810.0038.0180.0968.0000.1252  ;3080 C00100: D_RC[0D]   ;GET DATA READ
U 1252, 000D.0020.0180.0A20.0010.1253  ;3081  ALU_D[XOR]R[04],CLK.UBCC ;SEE DATA READ FROM CACHE NOT MEMORY
U 1253, 0000.013C.0180.0800.0000.10F0  ;3082  Z?
U 10F0, 0000.003D.0180.0800.0000.1183  ;3083 =0 ERROR1    ;DATA FROM CACHE IS BAD OR IT'S
          ;3084       ;  FROM MEMORY
          ;3085       ;============================
          ;3086       ;DATA: EXPECTED DATA
          ;3087       ; DATA READ
          ;3088       ;============================
U 10F1, 0010.0038.0180.0918.0200.1254  ;3089  VA_RC[3]   ;WRITE INVALIDATE,SEE MISS
U 1254, 0000.003C.0180.9000.0000.1255  ;3090  MCT/INVALIDATE
U 1255, 0000.003C.0180.0803.0000.1256  ;3091  VA_VA+4
U 1256, 0000.003C.0180.9000.0000.1257  ;3092  MCT/INVALIDATE
U 1257, 0010.0038.0180.0918.0200.1258  ;3093  VA_RC[3]   ;READ SEE MISS
U 1258, 0000.003C.0180.C800.0000.1259  ;3094  MCT/READ.P   ;READ MAIN MEMORY
U 1259, 0001.003C.0180.09E8.0000.1038  ;3095  RC[0D]_D   ;STORE DATA READ IN RC D
U 1038, 0000.003D.0180.0800.0000.108C  ;3096 =00 CALL[NOINTR]
U 1039, 0000.003D.0180.0800.0000.1183  ;3097  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 103A, 0000.003C.75F0.2C00.0000.125A  ;3098  Q_ID[MAINT]   ;SEE A CACHE MISS
U 125A, 000D.2034.0180.0A18.0010.125B  ;3099 = ALU_Q[AND]R[03],CLK.UBCC
U 125B, 0000.013C.0180.0800.0000.10F2  ;3100  Z?
U 10F2, 0000.003D.0180.0800.0000.1183  ;3101 =0 ERROR1    ;INVALIDATE FAILED, CACHE HIT
U 10F3, 0810.0038.0180.0968.0000.125C  ;3102  D_RC[0D]   ;CHECK DATA READ AFTER INVALIDATE
U 125C, 000D.0020.0180.0A28.0010.125D  ;3103  ALU_D[XOR]R[5],CLK.UBCC
U 125D, 0000.013C.0180.0800.0000.10F4  ;3104  Z?
U 10F4, 0000.003D.0180.0800.0000.1183  ;3105 =0 ERROR1    ;DATA READ ON MISS BAD,
          ;3106       ;  AFTER INVALIDATE
          ;3107       ;============================
          ;3108       ;DATA: EXPECTED DATA
          ;3109       ; DATA READ
          ;3110       ;============================
U 10F5, 0010.0038.0180.0960.0010.125E  ;3111  ALU_RC[0C],CLK.UBCC  ;GROUP FLAG SET ???
U 125E, 0000.013C.0180.0800.0000.10F6  ;3112  Z?
U 10F6, 0003.003C.0180.09E0.0000.125F  ;3113 =0 J/C00END,RC[0C]_0  ;YES END TESTING
U 10F7, 0018.0038.0580.09E0.0000.10D4  ;3114  RC[0C]_K[.1],J/C00020  ;SET IT AND REPEAT
U 125F, 0000.003C.0180.0800.0000.10F8  ;3115 C00END: NOP    ; ALL DONE.
          ;3116
```

```
;3117  .PAGE "TEST 1D0 QUAD WORD PLACEMENT IN CACHE ON BYTE READS"
;3118 ;******************************************************************************
;3119 ;++
;3120 ;
;3121 ; "CHET04.MIC"
;3122 ;
;3123 ; PLACEMENT OF DATA IN THE CACHE ON BYTE READS TO MEMORY
;3124 ;
;3125 ; TEST DESCRIPTION
;3126 ;
;3127 ; QUAD PLACENEMT OF DATA IN THE CACHE DURING BYTE READS TO
;3128 ; MAIN MEMORY. CHECKS TO SEE THAT THE DATA WAS PROPERLY PLACED
;3129 ; IN THE CHACE BY LONG WORD READ AND THEN CHECKING FOR HITS. THE
;3130 ; TEST CATCHES CACHE PARITY ERRORS ON HITS. THE TEST DATA USED
;3131 ; IS A BYTE OF ALL ONES FLOATED(SHIFTED BY BYTE POSITION) ACCROSS
;3132 ; A QUAD WORD OF ALL ZERO'S. THE BYTE READ ,USED TO FILL THE CACHE
;3133 ; IS MADE AT THE BASE OF MEMORY.
;3134 ;
;3135 ; SEQUENCE OF EVENTS
;3136 ; ==========================
;3137 ;
;3138 ; 1)INVALIDATE CACHE, BOTH LONG WORDS
;3139 ; 2)CACHE OFF ,SBI ON
;3140 ; 3)WRITE DATA TO MEMORY
;3141 ;  RA 2 => VA=(RC 3)  LOWER
;3142 ;  RA 3 => VA+4       UPPER
;3143 ; 4)CACHE ON
;3144 ; 5)READ BYTE AT THE BASE ADDRESS OF MEMORY
;3145 ;  A) SEE MISS
;3146 ;  B) ASSUME CORRECT DATA IS READ FROM MEMORY
;3147 ; 6)SBI OFF, CACHE ON
;3148 ; 7)READ LONG WORDS (BOTH )
;3149 ;  SEE: HIT
;3150 ;       CORRECT DATA
;3151 ;       HIT FROM SAME GROUP(AND ONLY ONE)
;3152 ;   EACH TIME
;3153 ;
;3154 ; LOGIC DESCRIPTION
;3155 ;
;3156 ; SBL(M8218) AND SBH(M8219)
;3157 ;  CACHE CONTROL LOGIC
;3158 ;  EXTENDED READ CONTROL
;3159 ;  MD BUS DRIVERS
;3160 ;  CACHE WRITE CONTROL LOGIC
;3161 ; CACHE PARITY CHECKERS
;3162 ; CACHE WRITE LOGIC
;3163 ;
;3164 ; ERROR LOGOUT
;3165 ;
;3166 ; DATA: UPPER EXPECTED DATA
;3167 ;  LOW EXPECTED DATA
;3168 ;  UPPER DATA READ
;3169 ;  LOW DATA READ
;3170 ;  OPTIONALLY USED(OPTIONALLY USED)
;3171 ;  OPTIONALLY USED(SEE LISTING)
```

```
            ;3172 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;3173 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;3174 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;3175 ;  INTERRUPT FLAG:
            ;3176 ;   0=>NO INTERRUPT OCCURRED
            ;3177 ;   1=>AN INTERRUPT OCCURRED
            ;3178 ;
            ;3179 ; SYNC POINT DESCRIPTION
            ;3180 ;
            ;3181 ;--
            ;3182 ;
            ;3183 ;=================================================================
            ;3184 ;  SCRATCH PAD REQUIREMENTS
            ;3185 ;
            ;3186 ; RA 0 BYTE DISPLACEMENT 0-7
            ;3187 ; RA 1 SHIFT COUNT TO GENERATE EXPECTED DATA
            ;3188 ; RA 2 ]
            ;3189 ; RA 3 ]EXPECTED DATA BUFFER
            ;3190 ; RA 4 SHUT OFF CACHE, TURN SBI ON  ^X18000
            ;3191 ; RA 5 CACHE ON, SBI OFF   ^X1000
            ;3192 ; RA 6 CACHE HIT MASK (FOR EITHER GROUP) ^X600
            ;3193 ; RA F U-TRAP FLAG
            ;3194 ;
            ;3195 ; RC 0 ADR OF CONFIG REG A   ^X2000 2000
            ;3196 ; RC 1 ADR OF CONFIG REG B   ^X2000 2004
            ;3197 ; RC 2 ADR OF CONFIG REG C   ^X2000 2008
            ;3198 ; RC 3 BASE ADDRESS OF MEMORY   ^X0
            ;3199 ; RC 9 ARG TO CONSOLE
            ;3200 ; RC A ARG TO CONSOLE
            ;3201 ; RC B ARG TO CONSOLE
            ;3202 ; RC C ARG TO CONSOLE
            ;3203 ; RC D ARG TO CONSOLE
            ;3204 ; RC E ARG TO CONSOLE
            ;3205 ; RC F # OF ARG TO CONSOLE
            ;3206 ;
            ;3207 ;*****************************************************************
            ;3208 ;
            ;3209 ;
            ;3210 =0
U 10F8,  0000.003D.0180.0800.0000.1053  ;3211 C04000: NEWTST
U 10F9,  0003.003C.0180.0AF8.0000.10FA  ;3212  R[0F]_0     ;CLEAR U-TRAP FLAG
U 10FA,  0000.003D.0180.0800.0000.11EE  ;3213 =0 CALL[CONFIG.MS780E]  ; Configure MS780E (if any)
U 10FB,  0818.0038.0DF8.0800.0000.1260  ;3214  D_K[.3],Q_0    ;RA 4
U 1260,  0000.003C.6180.0800.0084.7261  ;3215  SC_K[.F]
U 1261,  0D00.003C.0180.0800.0000.1262  ;3216  D_DAL.SC
U 1262,  0001.003C.0180.0AA0.0000.1263  ;3217  R[4]_D
U 1263,  0F00.003C.8580.0800.0084.7264  ;3218  D_0,SC_K[.C]    ;RA 5
U 1264,  0801.001C.0180.0AA8.0000.1265  ;3219  R[5]_ALU,D_D.ORNOT.MASK
U 1265,  0818.0038.D5F8.0800.0000.1266  ;3220  D_K[.6],Q_0    ;RA 6
U 1266,  0000.003C.0180.0800.0084.7267  ;3221  SC_K[.8]
U 1267,  0D00.003C.0180.0800.0000.1268  ;3222  D_DAL.SC
U 1268,  0001.003C.0180.0AB0.0000.1269  ;3223  R[6]_D
U 1269,  0003.003C.0180.0998.0000.126A  ;3224  RC[3]_0     ;RC 3
            ;3225  ;
            ;3226  ;=================================================================
```

```
                ;3227  ;
U 126A. 0003.003C.0180.0A80.0000.126B   ;3228  R[0]_0     ;SET INITIAL BYTE DISPLACEMENT
U 126B. 0003.003C.0180.0A88.0000.126C   ;3229  R[1]_0     ;SET INITIAL SHIFT COUNT
U 126C. 0018.0038.F580.09F8.0000.10FC   ;3230  RC[0F]_K[.A]
U 10FC. 0000.003D.0180.0800.0000.1182   ;3231 =0 ERLOOP
U 10FD. 0000.003C.0180.0800.0000.10FE   ;3232  NOP
                ;3233 =0
U 10FE. 0000.003D.0180.0800.0000.11A1   ;3234 C04020: CALL[INIT]   ;SET CPU FOR TEST
U 10FF. 0F00.003C.0180.0800.0000.126D   ;3235  D_0
U 126D. 0000.003C.7580.3C00.0000.126E   ;3236 = ID[MAINT]_D   ;ENABLE CACHE,SBI
U 126E. 0010.0038.0180.0918.0200.126F   ;3237  VA_RC[03]    ;INVALIDATE CACHE
U 126F. 0000.003C.0180.9000.0000.1270   ;3238  MCT/INVALIDATE
U 1270. 0000.003C.0180.0803.0000.1271   ;3239  VA_VA+4
U 1271. 0000.003C.0180.9000.0000.1272   ;3240  MCT/INVALIDATE
U 1272. 0800.003C.0180.0A20.0000.1273   ;3241  D_R[4]     ;CACHE OFF,SBI ON
U 1273. 0000.003C.7580.3C00.0000.1274   ;3242  ID[MAINT]_D
U 1274. 0818.0038.05F8.0800.0000.1275   ;3243  D_K[.1],Q_D    ;FIND EXPECTED DATA IN RA 2,3
U 1275. 0000.003C.0180.0A08.0082.1276   ;3244  SC_ALU,ALU_R[1]    ;SET SHIFT COUNT
U 1276. 0018.0038.1D80.0800.0010.1277   ;3245 C04030: ALU_SC,CLK.UBCC    ;TEST SHIFT COUNT = 0 ??
U 1277. 0000.013C.0180.0800.0000.110A   ;3246  Z?
                ;3247 =0 Q_Q.LEFT2,D_D.LEFT2,SI/ASHL, ;SHIT THE DATA
U 110A. 0100.003C.0908.0800.0084.B276   ;3248    SC_SC-K[.2],J/C04030 ;DECREMENT COUNT,REPEAT LOOP
U 110B. 0001.003C.0180.0A90.0000.1278   ;3249  R[2]_D     ;SET EXPECTED DATA,LOW
U 1278. 0001.003C.0180.09E8.0000.1279   ;3250  RC[0D]_D
U 1279. 0001.203C.0180.0A98.0000.127A   ;3251  R[3]_Q     ;SET EXPECTED DATA HI
U 127A. 0001.203C.0180.09F0.0000.127B   ;3252  RC[0E]_Q    ;
U 127B. 0800.003C.0180.0A10.0000.127C   ;3253  D_R[02]     ;WRITE EXPECTED DATA TO MEMORY
U 127C. 0010.0038.0180.0918.0200.127D   ;3254  VA_RC[3]    ; CACHE IS OFF
U 127D. 0000.003C.0180.A800.0000.1110   ;3255  MCT/WRITE.P    ;WRITE LOW DATA
U 1110. 0818.0039.D580.0800.0000.11BB   ;3256 =0 D_K[.6],CALL[WAIT.LOOP]
U 1111. 0000.003C.0180.0800.0000.103C   ;3257  NOP
U 103C. 0000.003D.0180.0800.0000.108C   ;3258 =00 CALL[NOINTR]
U 103D. 0000.003D.0180.0800.0000.1183   ;3259  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 103E. 0800.003C.0180.0A1B.0000.127E   ;3260  VA_VA+4,D_R[3]    ;ADDRESS ,DATA FOR UPPER
U 127E. 0000.003C.0180.A800.0000.1112   ;3261 = MCT/WRITE.P
U 1112. 0818.0039.D580.0800.0000.11BB   ;3262 =0 D_K[.6],CALL[WAIT.LOOP]
U 1113. 0000.003C.0180.0800.0000.1040   ;3263  NOP
U 1040. 0000.003D.0180.0800.0000.108C   ;3264 =00 CALL[NOINTR]
U 1041. 0000.003D.0180.0800.0000.1183   ;3265  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1042. 0F00.003C.0180.0800.0000.127F   ;3266  D_0     ;TURN CACHE ON READ A BYTE
U 127F. 0810.0038.7580.3D18.0000.1280   ;3267 = ID[MAINT]_D,D_RC[3]  ;LOAD ADDRESS FOR READ
U 1280. 0819.0014.5D80.0800.0200.1281   ;3268  VA_ALU,D_D+K[.7]
U 1281. 0000.803C.0180.C800.0000.1044   ;3269  MCT/READ.P,DT/BYTE  ;READ A BYTE, CACHE SHOULD FILL
U 1044. 0000.003D.0180.0800.0000.108C   ;3270 =00 CALL[NOINTR]
U 1045. 0000.003D.0180.0800.0000.1183   ;3271  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1046. 0000.003C.75F0.2C00.0000.1282   ;3272  Q_ID[MAINT]    ;CHECK FOR CACHE MISS
U 1282. 000D.2034.0180.0A30.0010.1283   ;3273 = ALU_Q[AND]R[6],CLK.UBCC  ;CHECK CACHE BITS
U 1283. 0000.013C.0180.0800.0000.1114   ;3274  Z?
U 1114. 0000.003D.0180.0800.0000.1183   ;3275 =0 ERROR1     ;CACHE HIT ,TEST FAILED
U 1115. 0800.003C.0180.0A28.0000.1284   ;3276  D_R[5]     ;SBI OFF, CACHE ON
U 1284. 0010.0038.7580.3D18.0200.1285   ;3277  ID[MAINT]_D,VA_RC[03]  ;LOAD ADDRESS FOR LOW READ
U 1285. 0F00.003C.0180.0800.0084.7286   ;3278  D_0,SC_K[.8]    ;CATCH CACHE PARITY ERROR
U 1286. 0801.001C.0180.0AF8.0000.1287   ;3279  R[0F]_ALU,D_D.ORNOT.MASK
U 1287. 0000.003C.0180.C800.0000.1288   ;3280  MCT/READ.P    ;READ LOW DATA,2ND REFERENCE
U 1288. 0001.003C.0180.09D8.0000.1116   ;3281  RC[0B]_D    ;RESULT TO RC B
```

```
U 1116, 0000.003D.F580.0800.0084.718B  ;3282 =0 SETRCF[.A]   ;# OF ARG TO CONSOLE
U 1117, 0000.003C.0180.0A78.0010.1289  ;3283  ALU_R[0F],CLK.UBCC
U 1289, 0003.013C.0180.0AF8.0000.1118  ;3284  Z?,R[0F]_0
U 111B, 0000.003C.0180.0800.0000.1048  ;3285 =0 J/C04070   ;NO PARITY ERROR
U 1119, 0000.003C.79F0.2C00.0000.128A  ;3286  Q_ID[PARITY]   ;CACHE PARITY ERROR OCCURED
U 128A, 0001.203C.0180.09D0.0000.128B  ;3287  RC[0A]_Q   ;SAVE THE IMAGE
U 128B, 0000.003D.0180.0800.0000.1183  ;3288  ERROR1    ;CACHE PARITY ERROR ON LOW REFERENCE
          ;3289       ;============================
          ;3290       ;DATA: UPPER EXPECTED DATA
          ;3291       ; LOW EXPECTED DATA
          ;3292       ; NOT USED
          ;3293       ; LOW DATA READ FROM CACHE
          ;3294       ; CACHE PARITY REG IMAGE
          ;3295       ; NOT USED
          ;3296       ;============================
          ;3297 =00
U 1048, 0000.003D.0180.0800.0000.108C  ;3298 C04070: CALL[NOINTR]
U 1049, 0000.003D.0180.0800.0000.1183  ;3299  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 104A, 0000.003C.75F0.2C00.0000.128C  ;3300  Q_ID[MAINT]   ;CHECK FOR HIT,SAVE THE GROUP
U 128C, 000D.2034.01C0.0A30.0010.128D  ;3301 = Q_ALU,ALU_Q[AND]R[6],CLK.UBCC ;CHECK CACHE BITS
U 128D, 0001.213C.0180.09C8.0000.111A  ;3302  Z?,RC[9]_Q   ;SAVE THE GROUP IT CAME FROM
U 111A, 0000.003C.0180.0800.0000.128E  ;3303 =0 J/C04075   ;CACHE HIT ,TEST PASSED
U 111B, 0000.003D.0180.0800.0000.1183  ;3304  ERROR1    ;CACHE MISS TEST FAILED
U 128E, 0010.0038.0180.0918.0200.128F  ;3305 C04075: VA_RC[03]   ;READ UPPER DATA,2ND REFERENCE
U 128F, 0F00.003C.0180.0800.0084.7290  ;3306  D_0,SC_K[.8]   ;CATCH CACHE PARITY ERROR
U 1290, 0801.001C.0180.0AF8.0000.1291  ;3307  R[0F]_ALU,D_D.ORNOT.MASK
U 1291, 0000.003C.0180.0803.0000.1292  ;3308  VA_VA+4
U 1292, 0000.003C.0180.C800.0000.1293  ;3309  MCT/READ.P
U 1293, 0001.003C.0180.09E0.0000.1294  ;3310  RC[0C]_D   ;SAVE THE DATA
U 1294, 0000.003C.0180.0A78.0010.1295  ;3311  ALU_R[0F],CLK.UBCC
U 1295, 0003.013C.0180.0AF8.0000.111C  ;3312  Z?,R[0F]_0
U 111C, 0000.003C.0180.0800.0000.1050  ;3313 =0 J/C04080   ;NO PARITY ERROR
U 111D, 0000.003C.79F0.2C00.0000.1296  ;3314  Q_ID[PARITY]   ;CACHE PARITY ERROR OCCURED
U 1296, 0001.203C.0180.09D0.0000.1297  ;3315  RC[0A]_Q   ;SAVE THE IMAGE
U 1297, 0000.003D.0180.0800.0000.1183  ;3316  ERROR1    ;CACHE PARITY ERROR ,UPPER REFERENCE
          ;3317       ;============================
          ;3318       ;DATA: UPPER EXPECTED DATA
          ;3319       ; LOW EXPECTED DATA
          ;3320       ; UPPER DATA READ FROM CACHE
          ;3321       ; LOW DATA READ FROM CACHE
          ;3322       ; CACHE PARITY REG IMAGE
          ;3323       ; NOT USED
          ;3324       ;============================
          ;3325 =00
U 1050, 0000.003D.0180.0800.0000.108C  ;3326 C04080: CALL[NOINTR]
U 1051, 0000.003D.0180.0800.0000.1183  ;3327  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1052, 0000.003C.75F0.2C00.0000.1298  ;3328  Q_ID[MAINT]   ;CHECK FOR HIT ON UPPER READ
U 1298, 000D.2034.01C0.0A30.0010.1299  ;3329 = Q_ALU,ALU_Q[AND]R[6],CLK.UBCC ;CHECK CACHE BITS
U 1299, 0001.213C.0180.09D0.0000.111E  ;3330  Z?,RC[0A]_Q   ;SAVE GROUP IT CAME FROM
U 111E, 0000.003C.0180.0800.0000.129A  ;3331 =0 J/C04090   ;CACHE HIT,TEST PASSED
U 111F, 0000.003D.0180.0800.0000.1183  ;3332  ERROR1    ;CACHE MISS ,TEST FAILED
U 129A, 0810.0038.0180.0958.0000.129B  ;3333 C04090: D_RC[0B]   ;CHECK THE DATA READ
U 129B, 000D.0020.0180.0A10.0010.129C  ;3334  ALU_D[XOR]R[2],CLK.UBCC  ;FIRST THE LOW DATA
U 129C, 0810.0138.0180.0960.0000.1120  ;3335  Z?,D_RC[0C]
U 1120, 0000.003D.0180.0800.0000.1183  ;3336 =0 ERROR1    ;LOW DATA READ DURING HIT IS BAD
```

```
U 1121, 000D.0020.0180.0A18.0010.129D  ;3337  ALU_D[XOR]R[3],CLK.UBCC  ;CHECK THE UPPER DATA
U 129D, 0000.013C.0180.0800.0000.1122  ;3338  Z?
U 1122, 0000.003D.0180.0800.0000.1183  ;3339 =0 ERROR1     ;UPPER DATA READ BAD
          ;3340       ;==============================
          ;3341       ;DATA: UPPER EXPECTED DATA
          ;3342       ; LOW EXPECTED DATA
          ;3343       ; UPPER DATA READ FROM CACHE
          ;3344       ; LOW DATA READ FROM CACHE
          ;3345       ; HIT BITS FROM UPPER REFERENCE
          ;3346       ; HIT BITS FROM LOW REFERENCE
          ;3347       ;==============================
U 1123, 0810.0038.0180.0948.0000.129E  ;3348  D_RC[9]     ;CHECK FOR HITS FROM ONLY ONE
U 129E, 000D.0020.0180.0A30.0010.129F  ;3349  ALU_D[XOR]R[6],CLK.UBCC  ; AND ONLY ONE GROUP
U 129F, 0000.013C.0180.0800.0000.1124  ;3350  Z?
U 1124, 0000.003C.0180.0800.0000.12A0  ;3351 =0 J/C04140    ;BOTH ARE NOT SET,CONTINUE
U 1125, 0000.003D.0180.0800.0000.1183  ;3352  ERROR1     ;BOTH GROUP BITS SET
          ;3353       ;==============================
          ;3354       ;DATA: UPPER EXPECTED DATA
          ;3355       ; LOW EXPECTED DATA
          ;3356       ; UPPER DATA READ FROM CACHE
          ;3357       ; LOW DATA READ FROM CACHE
          ;3358       ; HIT BITS FROM UPPER REFERENCE
          ;3359       ; HIT BITS FROM LOW REFERENCE
          ;3360       ;==============================
U 12A0, 0011.0020.0180.0950.0010.12A1  ;3361 C04140: ALU_D[XOR]RC[0A],CLK.UBCC ;BOTH REFERENCE GOT HITS FROM
U 12A1, 0000.013C.0180.0800.0000.1126  ;3362  Z?     ;  SAME GROUP ??
U 1126, 0000.003D.0180.0800.0000.1183  ;3363 =0 ERROR1     ;NO,EACH REFERENCE GOT HITS IN
          ;3364       ; A DIFFERENT GROUP
          ;3365       ;==============================
          ;3366       ;DATA: UPPER EXPECTED DATA
          ;3367       ; LOW EXPECTED DATA
          ;3368       ; UPPER DATA READ FROM CACHE
          ;3369       ; LOW DATA READ FROM CACHE
          ;3370       ; HIT BITS FROM UPPER REFERENCE
          ;3371       ; HIT BITS FROM LOW REFERENCE
          ;3372       ;==============================
U 1127, 0800.003C.0180.0A00.0000.12A2  ;3373  D_R[0]     ;CHECK BYTE DISPLACEMENT = 7
U 12A2, 0019.0020.5D80.0800.0010.12A3  ;3374  ALU_D.XOR.K[.7],CLK.UBCC
U 12A3, 0000.013C.0180.0800.0000.1128  ;3375  Z?
U 1128, 0000.003C.0180.0800.0000.12A4  ;3376 =0 J/C04160    ;NOT EQUAL TO 7
U 1129, 0000.003C.0180.0800.0000.12A8  ;3377  J/C04END    ; EQUAL TO SEVEN, END TEST
U 12A4, 0819.0014.0580.0A80.0000.12A5  ;3378 C04160: R[0]_ALU,D_D+K[.1]  ;INCREMENT BYTE DISPLACMENT
U 12A5, 0100.003C.0180.0800.0000.12A6  ;3379  D_D.LEFT2
U 12A6, 0500.003C.0180.0800.0000.12A7  ;3380  D_D.LEFT
U 12A7, 0001.003C.0180.0A88.0000.10FE  ;3381  R[1]_D,J/C04020    ;NEW SHIFT COUNT, REPEAT TEST
          ;3382
U 12A8, 0000.003C.0180.0800.0000.112A  ;3383 C04END: NOP     ; ALL DONE.
          ;3384
```

```
;3385  .PAGE "TEST 1D1 PRELIMINARY CACHE-MEMORY DATA TEST  "
;3386 ;================================================================================
;3387 ;**
;3388 ;
;3389 ; "CHET03.MIC"
;3390 ;
;3391 ; MEMORY-CACHE DATA TEST
;3392 ;
;3393 ; TEST DESCRIPTION
;3394 ;
;3395 ; 1) FLOATS A 1 THROUGH A FIELD OF ZERO'S
;3396 ;  AND ITS COMPLIMENT
;3397 ; 2) CHECKS TO SEE THAT THE DATA PATTERN APPEARS
;3398 ;     PROPERLY IN CACHE. DOES NOT CHECK
;3399 ;     WRITE HITS IN THE CACHE. CATCHES CACHE PARITY
;3400 ;     ERRORS WHEN THE CACHE IS ACCESSED DURING HITS.
;3401 ;
;3402 ; LOGIC DESCRIPTION
;3403 ;
;3404 ; CACHE CONTROL ON THE SBL(M8218) AND SBH(M8219) BOARDS.
;3405 ; MD BUS DRIVERS AND PARITY GENERATORS ON THE SBL,SBH
;3406 ;  BOARDS.
;3407 ; THE PARITY CHECKERS ON THE CACHE DATA MATRIX  ARE TESTED
;3408 ;  WHEN A CHECHE HIT OCCURES.
;3409 ; IF DATA BAD AND NO CACHE PARITY ERROS THE THE MEMORY MAY
;3410 ;  HAVE DRROPPED BITS, AS ECC IS INHIBITED.
;3411 ;
;3412 ; ERROR LOGOUT
;3413 ;
;3414 ; DATA: UPPER EXPECTED DATA
;3415 ;  LOW EXPECTED DATA
;3416 ;  UPPER DATA READ(HIT/MISS)
;3417 ;  LOW DATA READ(HIT/MISS)
;3418 ;  OPTIONALLY GIVEN
;3419 ;  UNDEFINED
;3420 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3421 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3422 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3423 ;  INTERRUPT FLAG:
;3424 ;   0=>NO INTERRUPT OCCURRED
;3425 ;   1=>AN INTERRUPT OCCURRED
;3426 ;
;3427 ; SYNC POINT DESCRIPTION
;3428 ;
;3429 ;--
;3430 ;****************************************************************
;3431 ;  SCRATCH PAD ASSIGNMENTS AND REQUIREMENTS
;3432 ;
;3433 ; RA,B  DESCRIPTION
;3434 ; ---- -------------------------------------------
;3435 ;  0 DATA LOWER 32 BITS
;3436 ;  1 DATA UPPER 32 BITS
;3437 ;  2 COMPLIMENT FLAG
;3438 ;  3 LOW WRITE DATA BUFFER
;3439 ;  4 UPPER WRITE DATA BUFFER
```

```
            ;3440 ;  5
            ;3441 ;  6
            ;3442 ;  7
            ;3443 ;  8
            ;3444 ;  9
            ;3445 ;  A
            ;3446 ;  B
            ;3447 ;  C
            ;3448 ;  D
            ;3449 ;  E
            ;3450 ;  F U-TRAP FLAG
            ;3451 ;
            ;3452 ; RC  DESCRIPTION
            ;3453 ; --  ----------------------------------------
            ;3454 ;  0 CONFIGURATION REG A ADDRESS ^X2000 2000
            ;3455 ;  1 CONFIGURATION REG B ADDRESS ^X2000 2004
            ;3456 ;  2 CONFIGURATION REG C ADDRESS ^X2000 2008
            ;3457 ;  3 BASE ADDRESS FOR MEMORY  ^X0
            ;3458 ;  4
            ;3459 ;  5 CONSTANT TO SHUT OFF CACHE ^X18000
            ;3460 ;  6 CONSTANT TO DISABLE SBI CYCLES ^X1000
            ;3461 ;  7 MASK FOR CACHE HIT BITS  ^X600
            ;3462 ;  A CACHE PARITY ERROR REGISTER IMAGE
            ;3463 ;  B LOW DATA READ
            ;3464 ;  C UPPER DATA READ
            ;3465 ;  D LOW EXPECTED DATA
            ;3466 ;  E UPPER EXPECTED DATA
            ;3467 ;  F # OF ARG TO CONSOLE  ^X4(5)
            ;3468 ;
            ;3469 ;*************************************************************
            ;3470 ;
            ;3471 =0
U 112A, 0000.003D.0180.0800.0000.1053  ;3472 C03000: NEWTST
U 112B, 0003.003C.0180.0AF8.0000.112C  ;3473  R[0F]_0     ;CLEAR U-TRAPS
U 112C, 0000.003D.0180.0800.0000.11EE  ;3474 =0 CALL[CONFIG.MS780E]  ; Configure MS780E (if any)
U 112D, 0003.003C.0180.0998.0000.12A9  ;3475  RC[3]_0     ;RC 3
U 12A9, 0818.003B.0DF8.0800.0000.12AB  ;3476  D_K[.3],Q_0    ;RC 5
U 12AB, 0000.003C.6180.0800.0084.72AC  ;3477  SC_K[.F]
U 12AC, 0D00.003C.0180.0800.0000.12AD  ;3478  D_DAL.SC
U 12AD, 0001.003C.0180.09A8.0000.12AE  ;3479  RC[5]_D
U 12AE, 0F00.003C.8580.0800.0084.72AF  ;3480  SC_K[.C],D_0    ;RC 6
U 12AF, 0801.001C.0180.09B0.0000.12B0  ;3481  RC[6]_ALU,D_D.ORNOT.MASK
U 12B0, 0818.003B.D180.0800.0000.12B1  ;3482  D_K[.C0]    ;RC 7
U 12B1, 0100.003C.0180.0800.0000.12B2  ;3483  D_D.LEFT2
U 12B2, 0500.003C.0180.0800.0000.12B3  ;3484  D_D.LEFT
U 12B3, 0001.003C.0180.09B8.0000.12B4  ;3485  RC[7]_D
            ;3486  ;
            ;3487  ;=============================================================
            ;3488  ;
U 12B4, 0003.003C.0180.0A90.0000.12B5  ;3489  R[2]_0     ;SET COMPLIMENT FLAG
U 12B5, 0018.0038.0580.0A80.0000.12B6  ;3490 C03005: R[0]_K[.1]   ;SET INITIAL DATA PATTERNS
U 12B6, 0003.003C.0180.0A88.0000.12B7  ;3491  R[1]_0
U 12B7, 0018.0038.F580.09F8.0000.112E  ;3492  RC[0F]_K[.A]
            ;3493 =0
U 112E, 0000.003D.0180.0800.0000.11A1  ;3494 C03010: CALL[INIT]
```

```
U 112F. 0F00.003C.0180.0800.0000.12B8  ;3495  D_0     ;ENABLE CACHE
U 12B8. 0000.003C.7580.3C00.0000.12B9  ;3496  ID[MAINT]_D
U 12B9. 0010.0038.0180.0918.0200.12BA  ;3497  VA_RC[03]
U 12BA. 0000.003C.0180.9000.0000.12BB  ;3498  MCT/INVALIDATE   ;INITIALIZE THE CACHE GROUPS
U 12BB. 0000.003C.0180.0803.0000.12BC  ;3499  VA_VA+4
U 12BC. 0000.003C.0180.9000.0000.12BD  ;3500  MCT/INVALIDATE
U 12BD. 0810.0038.0180.0928.0000.12BE  ;3501  D_RC[05]   ;ENABLE SBI,CACHE OFF
U 12BE. 0000.003C.7580.3C00.0000.12BF  ;3502  ID[MAINT]_D
U 12BF. 0010.0038.0180.0918.0200.12C0  ;3503  VA_RC[03]   ;WRITE LOWER DATA
U 12C0. 0000.003C.0180.0A10.0010.12C1  ;3504  ALU_R[2],CLK.UBCC  ;COMPLIMENT THE DATA ???
U 12C1. 0800.013C.0180.0A00.0000.1130  ;3505  Z?,D_R[0]
U 1130. 0801.0028.0180.0800.0000.1131  ;3506 =0 D_NOT.D
U 1131. 0001.003C.0180.0A98.0000.12C2  ;3507  R[3]_D
U 12C2. 0001.003C.0180.09E8.0000.12C3  ;3508  RC[0D]_D   ;SET EXPECTED DATA
U 12C3. 0000.003C.0180.A800.0000.1132  ;3509  MCT/WRITE.P   ; WRITE THE DATA
U 1132. 0818.0039.D580.0800.0000.11BB  ;3510 =0 D_K[.6],CALL[WAIT.LOOP]
U 1133. 0000.003C.0180.0800.0000.1054  ;3511  NOP
U 1054. 0000.003D.0180.0800.0000.108C  ;3512 =00 CALL[NOINTR]
U 1055. 0000.003D.0180.0800.0000.1183  ;3513  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1056. 0000.003C.0180.0A10.0010.12C4  ;3514  ALU_R[2],CLK.UBCC  ;COMPLIMENT UPPER DATA ??
U 12C4. 0800.013C.0180.0A0B.0000.1134  ;3515 = Z?,D_R[01],VA_VA+4
U 1134. 0801.0028.0180.0800.0000.1135  ;3516 =0 D_NOT.D     ;COMPLIMENT DATA
U 1135. 0001.003C.0180.0AA0.0000.12C5  ;3517  R[4]_D     ;SAVE IT IN THE BUFFER
U 12C5. 0001.003C.0180.09F0.0000.12C6  ;3518  RC[0E]_D   ;SET EXPECTED DATA TO CONSOLE
U 12C6. 0000.003C.0180.A800.0000.1136  ;3519  MCT/WRITE.P   ;WRITE UPPER DATA
U 1136. 0818.0039.D580.0800.0000.11BB  ;3520 =0 D_K[.6],CALL[WAIT.LOOP]
U 1137. 0000.003C.0180.0800.0000.1060  ;3521  NOP
U 1060. 0000.003D.0180.0800.0000.108C  ;3522 =00 CALL[NOINTR]
U 1061. 0000.003D.0180.0800.0000.1183  ;3523  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1062. 0F00.003C.0180.0800.0000.12C7  ;3524  D_0     ;ENABLE THE CACHE
U 12C7. 0000.003C.7580.3C00.0000.12C8  ;3525 = ID[MAINT]_D
U 12C8. 0010.0038.0180.0918.0200.12C9  ;3526  VA_RC[3]   ;READ LOW DATA FROM MEMORY
U 12C9. 0000.003C.0180.C800.0000.12CA  ;3527  MCT/READ.P
U 12CA. 0001.003C.0180.09D8.0000.1064  ;3528  RC[0B]_D   ;SAVE THE DATA READ
U 1064. 0000.003D.0180.0800.0000.108C  ;3529 =00 CALL[NOINTR]
U 1065. 0000.003D.0180.0800.0000.1183  ;3530  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1066. 0000.003C.75F0.2C00.0000.12CB  ;3531  Q_ID[MAINT]   ;CHECK FOR MISS ON LOW REFERENCE
U 12CB. 0011.2034.0180.0938.0010.12CC  ;3532 = ALU_Q[AND]RC[7],CLK.UBCC
U 12CC. 0000.013C.0180.0800.0000.1138  ;3533  Z?
U 1138. 0000.003D.0180.0800.0000.1183  ;3534 =0 ERROR1     ;CACHE HIT SET ON FIRST REFERENCE
U 1139. 0000.003C.0180.0803.0000.12CD  ;3535  VA_VA+4     ;READ UPPER WORD
U 12CD. 0F00.003C.0180.0800.0084.72CE  ;3536  D_0,SC_K[.8]   ;CATCH CACHE PARITY ERRORS
U 12CE. 0801.001C.0180.0AF8.0000.12CF  ;3537  R[0F]_ALU,D_D.ORNOT.MASK
U 12CF. 0000.003C.0180.C800.0000.12D0  ;3538  MCT/READ.P   ;REFRENCE TO SAME QUAD LOCATION
U 12D0. 0001.003C.0180.09E0.0000.12D1  ;3539  RC[0C]_D   ;SAVE UPPER DATA READ
U 12D1. 0000.003C.0180.0A78.0010.12D2  ;3540  ALU_R[0F],CLK.UBCC  ;CACHE PARITY U-TRAP ???
U 12D2. 0003.013C.0180.0AF8.0000.113A  ;3541  Z?,R[0F]_0
U 113A. 0000.003C.0180.0800.0000.1068  ;3542 =0 J/C03030   ;NO U-TRAP
U 113B. 0000.003C.79F0.2C00.0000.12D3  ;3543  Q_ID[PARITY]   ;YES , IT U-TRAPPED
        ;3544       ;  GET PARITY REGISTER
U 12D3. 0001.203C.0180.09D0.0000.12D4  ;3545  RC[0A]_Q
U 12D4. 0000.003D.0180.0800.0000.1183  ;3546  ERROR1     ;CACHE PARITY ERROR OCCURED
        ;3547       ;============================
        ;3548       ;DATA: UPPER EXPECTED DATA
        ;3549       ; LOW EXPECTED DATA
```

```
            ;3550       ; UPPER DATA READ(HIT)
            ;3551       ; LOW DATA READ(MISS)
            ;3552       ; IMAGE OF CACHE PARITY REG
            ;3553       ;=============================
            ;3554 =00
U 1068. 0000.003D.0180.0800.0000.108C  ;3555 C03030: CALL[NOINTR]
U 1069. 0000.003D.0180.0800.0000.1183  ;3556  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 106A. 0000.003C.75F0.2C00.0000.12D5  ;3557  Q_ID[MAINT]    ;CHECK FOR HIT,FROM THE SECOND
U 12D5. 0011.2034.0180.0938.0010.12D6  ;3558 = ALU_Q[AND]RC[7],CLK.UBCC ;  REFERENCE
U 12D6. 0000.013C.0180.0800.0000.113C  ;3559  Z?
U 113C. 0000.003C.0180.0800.0000.12D7  ;3560 =0 J/C03040    ;CACHE  HIT
U 113D. 0000.003D.0180.0800.0000.1183  ;3561  ERROR1     ;CACHE MISS ON SECOND REFERENCE
U 12D7. 0810.0038.0180.0968.0000.12D8  ;3562 C03040: D_RC[0D]    ;CHECK LOW DATA READ DURING MISS
U 12D8. 000D.0020.0180.0A18.0010.12D9  ;3563  ALU_D[XOR]R[3],CLK.UBCC
U 12D9. 0800.013C.0180.0A20.0000.113E  ;3564  Z?,D_R[4]    ;GET EXPECTED UPPER DATA
U 113E. 0000.003D.0180.0800.0000.1183  ;3565 =0 ERROR1     ;LOW RESULT BAD,ON MISS
U 113F. 0011.0020.0180.0960.0010.12DA  ;3566  ALU_D[XOR]RC[0C],CLK.UBCC ;CHECK UPPER RESULT
U 12DA. 0000.013C.0180.0800.0000.1140  ;3567  Z?     ;BRANCH ON UPPER RESULT
U 1140. 0000.003D.0180.0800.0000.1183  ;3568 =0 ERROR1     ;UPPER DATA BAD,ON HIT
            ;3569       ;=============================
            ;3570       ;DATA: UPPER EXPECTED DATA
            ;3571       ; LOW EXPECTED DATA
            ;3572       ; UPPER DATA READ(HIT)
            ;3573       ; LOW DATA READ(MISS)
            ;3574       ;=============================
            ;3575   ;=============================================================
            ;3576   ;
            ;3577   ;SHUT OFF SBI CYCLES, READ JUST FROM CACHE
            ;3578   ;
            ;3579   ;=============================================================
            ;3580   ;
U 1141. 0810.0038.0180.0930.0000.12DB  ;3581  D_RC[06]    ;DISABLE SBI CYCLES
U 12DB. 0000.003C.7580.3C00.0000.12DC  ;3582  ID[MAINT]_D
U 12DC. 0010.0038.0180.0918.0200.12DD  ;3583  VA_RC[3]    ;READ LOW WORD
U 12DD. 0F00.003C.0180.0800.0084.72DE  ;3584  D_0,SC_K[.8]    ;CATCH PARITY ERRORS(CACHE)
U 12DE. 0801.001C.0180.0AF8.0000.12DF  ;3585  R[0F]_ALU,D_D.ORNOT.MASK
U 12DF. 0000.003C.0180.C800.0000.12E0  ;3586  MCT/READ.P
U 12E0. 0001.003C.0180.09D8.0000.12E1  ;3587  RC[0B]_D    ;SAVE THE DATA READ
U 12E1. 0000.003C.0180.0A78.0010.12E2  ;3588  ALU_R[0F],CLK.UBCC  ;CACHE PARITY U-TRAP ???
U 12E2. 0003.013C.0180.0AF8.0000.1142  ;3589  Z?,R[0F]_0
U 1142. 0000.003C.0180.0800.0000.106C  ;3590 =0 J/C03050    ;NO U-TRAP
U 1143. 0000.003C.79F0.2C00.0000.12E3  ;3591  Q_ID[PARITY]    ;GET PARITY REGISTER
U 12E3. 0001.203C.0180.09D0.0000.12E4  ;3592  RC[0A]_Q
U 12E4. 0000.003D.0180.0800.0000.1183  ;3593  ERROR1     ;CACHE PARITY ERROR OCCURED
            ;3594       ;=============================
            ;3595       ;DATA: UPPER EXPECTED DATA
            ;3596       ; LOW EXPECTED DATA
            ;3597       ; UPPER DATA READ(HIT)
            ;3598       ; LOW DATA READ(HIT)
            ;3599       ; CACHE PARITY ERROR REG IMAGE
            ;3600        ;=============================
            ;3601 =00
U 106C. 0000.003D.0180.0800.0000.108C  ;3602 C03050: CALL[NOINTR]
U 106D. 0000.003D.0180.0800.0000.1183  ;3603  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 106E. 0000.003C.75F0.2C00.0000.12E5  ;3604  Q_ID[MAINT]    ;CHECK FOR CACHE HIT
```

```
U 12E5, 0011,2034,0180,0938,0010,12E6  ;3605 = ALU_Q[AND]RC[7],CLK.UBCC ;  SBI CYCLES ARE DISABLED
U 12E6, 0000,013C,0180,0800,0000,1144  ;3606  Z?
U 1144, 0000,003C,0180,0800,0000,12E7  ;3607 =0 J/C03080   ;CACHE HIT
U 1145, 0000,003D,0180,0800,0000,1183  ;3608  ERROR1    ;CACHE MISS ,SBI CYCLES OFF
U 12E7, 0000,003C,0180,0803,0000,12E8  ;3609 C03080: VA_VA+4    ;READ UPPER DATA
U 12E8, 0F00,003C,0180,0800,0084,72E9  ;3610  D_0,SC_K[.8]   ;CATCH CACHE PARITY ERRORS
U 12E9, 0801,001C,0180,0AF8,0000,12EA  ;3611  R[0F]_ALU,D_D.ORNOT.MASK
U 12EA, 0000,003C,0180,C800,0000,12EB  ;3612  MCT/READ.P
U 12EB, 0001,003C,0180,09E0,0000,12EC  ;3613  RC[0C]_D
U 12EC, 0000,003C,0180,0A78,0010,12ED  ;3614  ALU_R[0F],CLK.UBCC  ;CACHE PARITY U-TRAP ???
U 12ED, 0003,013C,0180,0AF8,0000,1146  ;3615  Z?,R[0F]_0
U 1146, 0000,003C,0180,0800,0000,12F0  ;3616 =0 J/C03090   ;NO U-TRAP
U 1147, 0000,003C,79F0,2C00,0000,12EE  ;3617  Q_ID[PARITY]   ;GET PARITY REGISTER
U 12EE, 0001,203C,0180,09D0,0000,12EF  ;3618  RC[0A]_Q
U 12EF, 0000,003D,0180,0800,0000,1183  ;3619  ERROR1    ;CACHE PARITY ERROR OCCURED
          ;3620       ;============================
          ;3621       ;DATA: UPPER EXPECTED DATA
          ;3622       ; LOW EXPECTED DATA
          ;3623       ; UPPER DATA READ(HIT)
          ;3624       ; LOW DATA READ(HIT)
          ;3625       ; CACHE PARITY ERROR REG IMAGE
          ;3626       ;============================
U 12F0, 0000,003C,75F0,2C00,0000,12F1  ;3627 C03090: Q_ID[MAINT]   ;CHECK FOR HIT UPPER REFERENCE
U 12F1, 0011,2034,0180,0938,0010,12F2  ;3628  ALU_Q[AND]RC[7],CLK.UBCC
U 12F2, 0000,013C,0180,0800,0000,1148  ;3629  Z?
U 1148, 0000,003C,0180,0800,0000,12F3  ;3630 =0 J/C03100   ;CACHE  HIT
U 1149, 0000,003D,0180,0800,0000,1183  ;3631  ERROR1    ;CACHE MISS, SBI WAS DISABLED
U 12F3, 0800,003C,0180,0A18,0000,12F4  ;3632 C03100: D_R[3]    ;GET LOW EXPECTED DATA
U 12F4, 0011,0020,0180,0958,0010,12F5  ;3633  ALU_D[XOR]RC[0B],CLK.UBCC ;CHECK UPPER DATA
U 12F5, 0800,013C,0180,0A20,0000,114A  ;3634  Z?,D_R[4]   ;SET UPPER EXPECTED DATA
U 114A, 0000,003D,0180,0800,0000,1183  ;3635 =0 ERROR1    ;LOW DATA BAD,
          ;3636       ;    CACHE HIT,SBI DIABLED
U 114B, 0011,0020,0180,0960,0010,12F6  ;3637  ALU_D[XOR]RC[0C],CLK.UBCC ;CHECK UPPER DATA
U 12F6, 0000,013C,0180,0800,0000,114C  ;3638  Z?    ;BRANCH ON UPPER RESULT
U 114C, 0000,003D,0180,0800,0000,1183  ;3639 =0 ERROR1    ;UPPER DATA BAD,
          ;3640       ;    SBI DISABLED,CACHE HIT
          ;3641       ;============================
          ;3642       ;DATA: UPPER EXPECTED DATA
          ;3643       ; LOW EXPECTED DATA
          ;3644       ; UPPER DATA READ(HIT)
          ;3645       ; LOW DATA READ(HIT)
          ;3646       ;============================
U 114D, 0000,003C,0180,0A08,0010,12F7  ;3647  ALU_R[1],CLK.UBCC  ;CHECK FOR LAST PATTERN
U 12F7, 0000,1B3C,0180,0800,0000,1017  ;3648  ALU.N?    ;BIT 31 SET ??
U 1017, 0000,003C,0180,0800,0000,12F8  ;3649 =0111 J/C03180   ;NO, SHIFT PATTERN AND REPEAT
U 101F, 0000,003C,0180,0800,0000,12FD  ;3650  J/C03190   ;YES,GO CHECK COMPLIMENT FLAG
U 12F8, 0800,003C,0180,0A00,0000,12F9  ;3651 C03180: D_R[0]    ;SHIFT THE DATA
U 12F9, 0000,003C,01C0,0A08,0000,12FA  ;3652  Q_R[1]
U 12FA, 0500,003C,0128,0800,0000,12FB  ;3653  D_D.LEFT,Q_Q.LEFT,SI/ASHL
U 12FB, 0001,003C,0180,0A80,0000,12FC  ;3654  R[0]_D
U 12FC, 0001,203C,0180,0A88,0000,112E  ;3655  R[1]_Q,J/C03010   ;GO REPEAT LOOP
U 12FD, 0000,003C,0180,0A10,0010,12FE  ;3656 C03190: ALU_R[2],CLK.UBCC  ;CHECK COMPLIMENT FLAG
U 12FE, 0000,013C,0180,0800,0000,114E  ;3657  Z?
U 114E, 0000,003C,0180,0800,0000,12FF  ;3658 =0 J/C03END   ;IT IS SET, END TESTING THIS TR
U 114F, 0018,0038,0580,0A90,0000,12B5  ;3659  R[2]_K[.1],J/C03005  ;SET IT,REPEAT TEST
```

```
U 12FF, 0000,003C,0180,0800,0000,1150  ;3660 C03END: NOP    ; ALL DONE.
            ;3661
```

```
;3662  .PAGE "TEST 1D2 CACHE UPDATE ON WRITE (BYTE ) HITS "
;3663 ;*****************************************************************************
;3664 ;**
;3665 ;
;3666 ; "CHET05.MIC"
;3667 ;
;3668 ; WRITE UPDATE ON CACHE HITS
;3669 ;
;3670 ; TEST DESCRIPTION
;3671 ;
;3672 ; DOES BYTE WRITE TO MEMORY WITH KNOWN HITS IN THE CACHE.
;3673 ; CHECKS TO SEE THAT BOTH THE CACHE AND MAIN MEMORY
;3674 ; ARE UPDATED. CHECKS ALL BYTE ADDRESS WITHIN THE
;3675 ; QUAD WORD. BOTH MEMORY AND CACHE WILL BE CHECKED TO SEE
;3676 ; THE DATA WAS WRITTEN. THE WRITE LOGIC WILL BE CHECKED TO
;3677 ; SEE THAT THE BYTE WRITE LINES ARE INDEPENDENT. THE DATA USED
;3678 ; FOR THE WRITE WILL BE BOTH ^X0000 00FF AND ^XFFFF FFFF.
;3679 ;
;3680 ; SEQUENCE OF EVENTS
;3681 ; ====================================
;3682 ;
;3683 ; 1)INVALIDATE QUAD LOCATION
;3684 ; 2)SHUT CACHE OFF, ENABLE SBI CYCLES
;3685 ; 3)CLEAR QUAD LOCATION,2 LONG WRITES TO MEMORY
;3686 ; 4)TURN CACHE ON
;3687 ; 5)READ UPPER LONG WORD
;3688 ;  SEE MISS
;3689 ; 6)READ LOWER LONG WORD
;3690 ; 7)BYTE WRITE WTHIN QUAD WORD
;3691 ;  RC 3 *0=>7
;3692 ;  USE: ^X 0000 00FF (PASS 1)
;3693 ;    ^X FFFF FFFF (PASS 2)
;3694 ; 8)SBI CYCLES OFF, CACHE ON
;3695 ; 9)READ BOTH LONG WORDS
;3696 ;  SEE HIT
;3697 ;  SEE MODIFIED DATA(COMPARE TO RA 3,4)
;3698 ; 10)SBI CYCLES ON,CACHE OFF
;3699 ; 11)READ BOTH LONG WORD CHECK FOR DATA AS IN
;3700 ;  9),I.E.COMPARE TO RA 3,4
;3701 ;
;3702 ; LOGIC DESCRIPTION
;3703 ;
;3704 ; SBL(M8218) AND SBH(M8219) FOR CACHE CONTROL LOGIC
;3705 ;  AND SBI WRITE CYCLES
;3706 ; CACHE ADDRESS MATRIX FOR DETEDTION OF HITS
;3707 ;
;3708 ; ERROR LOGOUT
;3709 ;
;3710 ; DATA: EXPECTED UPPER DATA
;3711 ;  EXPECTED LOW DATA
;3712 ;  UPPER DATA READ
;3713 ;  LOW DATA READ
;3714 ;  IMAGE OF D REG THAT WAS WRITTEN
;3715 ;  BYTE DISPLACEMENT USED FOR WRITE
;3716 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
```

```
;3717 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3718 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3719 ;  INTERRUPT FLAG:
;3720 ;   0=>NO INTERRUPT OCCURRED
;3721 ;   1=>AN INTERRUPT OCCURRED
;3722 ;
;3723 ; SYNC POINT DESCRIPTION
;3724 ;
;3725 ;--
;3726 ;
;3727 ;=================================================================
;3728 ;  SCRATCH PAD REQUIREMENTS
;3729 ;
;3730 ;
;3731 ; RA 0 BYTE DISPLACEMENT   0-7
;3732 ; RA 1 SHIFT COUNT
;3733 ; RA 2 DATA FLAG D REG IMAGE FOR WRITE
;3734 ;     =0: 00 00 00 FF
;3735 ;     =1: FF FF FF FF
;3736 ; RA 3 ]
;3737 ; RA 4 ]EXPECTED DATA BUFFER
;3738 ; RA 5 CONSTANT TO SHUT CACHE OFF,SBI ON ^X18000
;3739 ; RA 6 CONSTANT FOR HIT MASK   ^X600
;3740 ; RA 7 SBI OFF,CACHE ON CONSTANT  ^X1000
;3741 ; RA F U-TRAP FLAG
;3742 ;
;3743 ;
;3744 ;
;3745 ; RC 0 ADR OF CONFIG REG A   ^X2000 2000
;3746 ; RC 1 ADR OF CONFIG REG B   ^X2000 2004
;3747 ; RC 2 ADR OF CONFIG REG C   ^X2000 2008
;3748 ; RC 3 BASE ADDRESS OF MEMORY   ^X0
;3749 ; RC 9 BYTE DISPLACEMENT
;3750 ; RC A IMAGE OF D REG USED FOR WRITE
;3751 ; RC B LOW DATA READ
;3752 ; RC C UPPER DATA READ
;3753 ; RC D LOW EXPECTED DATA
;3754 ; RC E UPPER EXPECTED DATA
;3755 ; RC F # OF ARG TO CONSOLE   ^X6
;3756 ;
;3757 ;
;3758 ;*****************************************************************
;3759 ;
;3760 ;
;3761 ;
;3762 ;
;3763 =0
U 1150, 0000.003D,0180.0800,0000,1053  ;3764 C05000: NEWTST
U 1151, 0003.003C,0180.0AF8,0000,1152  ;3765  R[0F]_0    ;CLEAR U-TRAPS
U 1152, 0000.003D,0180.0800,0000,11EE  ;3766 =0 CALL[CONFIG.MS780E]  ; Configure MS780E (if any)
U 1153, 0818.0038,0DF8.0800,0000,1300  ;3767  D_K[.3],Q_0   ;RA 5
U 1300, 0000.003C,6180.0800,0084,7301  ;3768  SC_K[.F]
U 1301, 0D00.003C,0180.0800,0000,1302  ;3769  D_DAL.SC
U 1302, 0001.003C,0180.0AA8,0000,1303  ;3770  R[5]_D
U 1303, 081B.0038,D180.0800,0000,1304  ;3771  D_K[.C0]   ;RA 6
```

```
U 1304. 0100.003C.0180.0800.0000.1305  ;3772  D_D.LEFT2
U 1305. 0500.003C.0180.0800.0000.1306  ;3773  D_D.LEFT
U 1306. 0001.003C.0180.0AB0.0000.1307  ;3774  R[6]_D
U 1307. 0F00.003C.8580.0800.0084.7308  ;3775  SC_K[.C],D_0   ;RA 7
U 1308. 0801.001C.0180.0AB8.0000.1309  ;3776  R[7]_ALU,D_D.ORNOT.MASK
U 1309. 0003.003C.0180.0998.0000.1156  ;3777  RC[3]_0    ;RC 3
          ;3778  ;=====================================================
          ;3779  ;
U 1156. 0000.003D.F580.0800.0084.718B  ;3780 =0 SETRCF[.A]
U 1157. 0003.003C.0180.0A90.0000.130A  ;3781  R[2]_0    ;SET DATA FLAG
U 130A. 0003.003C.0180.0A80.0000.130B  ;3782 C05010: R[0]_0    ;SET INITIAL BYTE DISPLACEMENT
U 130B. 0003.003C.0180.0A88.0000.1158  ;3783  R[1]_0    ;SET INITIAL SHIFT COUNT
U 1158. 0000.003D.0180.0800.0000.1182  ;3784 =0 ERLOOP
U 1159. 0000.003C.0180.0800.0000.115A  ;3785 C05020: NOP
U 115A. 0000.003D.0180.0800.0000.11A1  ;3786 =0 CALL[INIT]
U 115B. 0F00.003C.0180.0800.0000.130C  ;3787  D_D    ;INVALIDATE THE CACHE
U 130C. 0000.003C.7580.3C00.0000.130D  ;3788  ID[MAINT]_D
U 130D. 0F10.0038.0180.0918.0200.130E  ;3789  D_0,VA_RC[03]   ;LOW WORD
U 130E. 0000.003C.0180.9000.0000.130F  ;3790  MCT/INVALIDATE
U 130F. 0F00.003C.0180.0803.0000.1310  ;3791  VA_VA+4,D_D
U 1310. 0000.003C.0180.9000.0000.1311  ;3792  MCT/INVALIDATE   ;UPPER WORD
U 1311. 0800.003C.0180.0A28.0000.1312  ;3793  D_R[05]    ;SHUT CACHE OFF
U 1312. 0000.003C.7580.3C00.0000.1313  ;3794  ID[MAINT]_D
U 1313. 0F10.0038.0180.0918.0200.1314  ;3795  VA_RC[3],D_0   ;CLEAR MEMORY LOCATIONS FOR TEST
U 1314. 0000.003C.0180.A800.0000.115C  ;3796  MCT/WRITE.P   ;WRITE LOW TEST DATA
U 115C. 0818.0039.D580.0800.0000.11BB  ;3797 =0 D_K[.6],CALL[WAIT.LOOP]
U 115D. 0000.003C.0180.0800.0000.1070  ;3798  NOP
U 1070. 0000.003D.0180.0800.0000.108C  ;3799 =00 CALL[NOINTR]
U 1071. 0000.003D.0180.0800.0000.1183  ;3800  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1072. 0F00.003C.0180.0803.0000.1315  ;3801  VA_VA+4,D_0   ;WRITE UPPER TEST DATA
U 1315. 0000.003C.0180.A800.0000.115E  ;3802 = MCT/WRITE.P
U 115E. 0818.0039.D580.0800.0000.11BB  ;3803 =0 D_K[.6],CALL[WAIT.LOOP]
U 115F. 0000.003C.0180.0800.0000.1074  ;3804  NOP
U 1074. 0000.003D.0180.0800.0000.108C  ;3805 =00 CALL[NOINTR]
U 1075. 0000.003D.0180.0800.0000.1183  ;3806  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1076. 0F00.003C.0180.0800.0000.1316  ;3807  D_D    ;TURN CACHE ON
U 1316. 0010.0038.7580.3D18.0200.1317  ;3808 = ID[MAINT]_D,VA_RC[03]  ;LOAD ADDRESS
U 1317. 0000.003C.0180.0803.0000.1318  ;3809  VA_VA+4    ;SET UPPER ADDRESS
U 1318. 0000.803C.0180.C800.0000.1078  ;3810  MCT/READ.P,BYTE   ;READ MEMORY,NO DATA
          ;3811        ;    TO BE CHECKED NOW
U 1078. 0000.003D.0180.0800.0000.108C  ;3812 =00 CALL[NOINTR]
U 1079. 0000.003D.0180.0800.0000.1183  ;3813  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 107A. 0000.003C.75F0.2C00.0000.1319  ;3814  Q_ID[MAINT]   ;CHECK FOR, MISS ON FIRST READ
U 1319. 000D.2034.0180.0A30.0010.131A  ;3815 = ALU_Q[AND]R[6],CLK.UBCC
U 131A. 0000.013C.0180.0800.0000.1160  ;3816  Z?
U 1160. 0000.003D.0180.0800.0000.1183  ;3817 =0 ERROR1    ;CACHE HIT ON FIST REFREFENCE
U 1161. 0010.0038.0180.0918.0200.131B  ;3818  VA_RC[03]   ;2ND REFERENCE SEE A HIT
U 131B. 0000.003C.0180.C800.0000.107C  ;3819  MCT/READ.P   ;WON'T CHECK THE DATA
U 107C. 0000.003D.0180.0800.0000.108C  ;3820 =00 CALL[NOINTR]
U 107D. 0000.003D.0180.0800.0000.1183  ;3821  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 107E. 0000.003C.75F0.2C00.0000.131C  ;3822  Q_ID[MAINT]   ;CHECK FOR HIT
U 131C. 000D.2034.0180.0A30.0010.131D  ;3823 = ALU_Q[AND]R[6],CLK.UBCC
U 131D. 0000.013C.0180.0800.0000.1162  ;3824  Z?
U 1162. 0000.003C.0180.0800.0000.131E  ;3825 =0 J/C05050   ;CAHE HIT
U 1163. 0000.003D.0180.0800.0000.1183  ;3826  ERROR1    ;CACHE MISS ON SECOND REFRENCE
```

```
U 131E. 0818.0038.49F8.0800.0000.131F  ;3827 C05050: D_K[.FF],Q_0   ;GENERATE THE EXPECTED DATA
U 131F. 0000.003C.0180.0A08.0082.1320  ;3828  SC_ALU,ALU_R[1]   ;GET SHIFT COUNT
U 1320. 0018.0038.1D80.0800.0010.1321  ;3829 C05060: ALU_SC,CLK.UBCC   ;SC = 0 ????
U 1321. 0000.013C.0180.0800.0000.1164  ;3830  Z?
        ;3831 =0 Q_Q.LEFT2,D_D.LEFT2,SI/ASHL, ;SHIFT THE DATA
U 1164. 0100.003C.0908.0800.0084.B320  ;3832   SC_SC-K[.2],J/C05060 ;ADJUST COUNT.LOOP
U 1165. 0001.003C.0180.09E8.0000.1322  ;3833  RC[0D]_D   ;SET EXPECTED DATA LOWER
U 1322. 0001.003C.0180.0A98.0000.1323  ;3834  R[3]_D
U 1323. 0001.203C.0180.09F0.0000.1324  ;3835  RC[0E]_Q   ;SET ECPECTED DATA UPPER
U 1324. 0001.203C.0180.0AA0.0000.1325  ;3836  R[4]_Q
U 1325. 0800.003C.0180.0A00.0000.1326  ;3837  D_R[0]    ;FIND THE TEST BYTE ADDRESS
U 1326. 0001.003C.0180.09C8.0000.1327  ;3838  RC[9]_D    ;SET DISPLACEMENT USED
U 1327. 0010.0038.01C0.0918.0000.1328  ;3839  Q_RC[3]    ;GET BASE ADDRESS OF MEMORY
U 1328. 001D.0014.0180.0800.0200.1329  ;3840  VA_ALU,ALU_D[A+B]Q  ;SET ADDRESS FOR WRITE
U 1329. 0000.003C.0180.0A10.0010.132A  ;3841  ALU_R[2],CLK.UBCC  ;CHECK THE DATA FLAG ??
U 132A. 0818.0138.4980.0800.0000.1166  ;3842  Z?,D_K[.FF]   ;SET DATA FOR = 0
U 1166. 081B.0000.0580.0800.0000.1167  ;3843 =0 D_0-K[.1]   ;FLAG = 1, DATA = FFFF FFFF
U 1167. 0001.003C.0180.09D0.0000.132B  ;3844  RC[0A]_D   ;SET DATA WRITTEN TO CONSOLE
U 132B. 0000.803C.0180.A800.0000.1168  ;3845  MCT/WRITE.P,DT/BYTE  ;WRITE THE DATA
U 1168. 0818.0039.D580.0800.0000.11BB  ;3846 =0 D_K[.6],CALL[WAIT.LOOP]
U 1169. 0000.003C.0180.0800.0000.1080  ;3847  NOP
U 1080. 0000.003D.0180.0800.0000.108C  ;3848 =00 CALL[NOINTR]
U 1081. 0000.003D.0180.0800.0000.1183  ;3849  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
        ;3850  ;
        ;3851  ;*****************************************
        ;3852  ;
        ;3853  ; DATA FROM THE CACHE TEST
        ;3854  ;
        ;3855  ;*****************************************
        ;3856  ;
U 1082. 0800.003C.0180.0A38.0000.132C  ;3857  D_R[7]    ;SBI OFF,CACHE ON
U 132C. 0000.003C.7580.3C00.0000.132D  ;3858 = ID[MAINT]_D
U 132D. 0F10.0038.0180.0918.0200.132E  ;3859  VA_RC[03],D_0   ;READ LOW CACHE DATA
U 132E. 0000.003C.0180.C800.0000.132F  ;3860  MCT/READ.P
U 132F. 0001.003C.0180.09D8.0000.1330  ;3861  RC[0B]_D
U 1330. 0000.003C.75F0.2C00.0000.1331  ;3862  Q_ID[MAINT]   ;CHECK FOR HIT
U 1331. 000D.2034.0180.0A30.0010.1332  ;3863  ALU_Q[AND]R[6],CLK.UBCC
U 1332. 0000.013C.0180.0800.0000.116A  ;3864  Z?
U 116A. 0000.003C.0180.0800.0000.1333  ;3865 =0 J/C05070
U 116B. 0000.003D.0180.0800.0000.1183  ;3866  ERROR1    ;CACHE MISS WHEN SBI DISABLED
U 1333. 0000.003C.0180.0803.0000.1334  ;3867 C05070: VA_VA+4    ;READ THE UPPER DATA
U 1334. 0000.003C.0180.C800.0000.1335  ;3868  MCT/READ.P
U 1335. 0001.003C.0180.09E0.0000.1336  ;3869  RC[0C]_D
U 1336. 0000.003C.75F0.2C00.0000.1337  ;3870  Q_ID[MAINT]   ;CHECK FOR HIT
U 1337. 000D.2034.0180.0A30.0010.1338  ;3871  ALU_Q[AND]R[6],CLK.UBCC
U 1338. 0000.013C.0180.0800.0000.116C  ;3872  Z?
U 116C. 0000.003C.0180.0800.0000.1339  ;3873 =0 J/C05090   ;CACHE HIT
U 116D. 0000.003D.0180.0800.0000.1183  ;3874  ERROR1    ;CACHE MISS WHEN SBI DISABLED
U 1339. 0800.003C.0180.0A18.0000.133A  ;3875 C05090: D_R[3]    ;CHECK LOW DATA READ FROM CACHE
U 133A. 0011.0020.0180.0958.0010.133B  ;3876  ALU_D[XOR]RC[0B],CLK.UBCC
U 133B. 0800.013C.0180.0A20.0000.116E  ;3877  Z?,D_R[4]
U 116E. 0000.003D.0180.0800.0000.1183  ;3878 =0 ERROR1    ;LOW DATA READ FROM CACHE BAD
U 116F. 0011.0020.0180.0960.0010.133C  ;3879  ALU_D[XOR]RC[0C],CLK.UBCC
U 133C. 0000.013C.0180.0800.0000.1170  ;3880  Z?
U 1170. 0000.003D.0180.0800.0000.1183  ;3881 =0 ERROR1    ;UPPER DATA READ FROM CACHE BAD
```

```
            ;3882       ;==============================
            ;3883       ;DATA: EXPECTED UPPER DATA
            ;3884       ; EXPECTED LOW DATA
            ;3885       ; UPPER DATA READ
            ;3886       ; LOW DATA READ
            ;3887       ; D REG IMAGE USED FOR WRITE
            ;3888       ; BYTE DISPLACEMENT USED
            ;3889       ;==============================
            ;3890   ;****************************************
            ;3891   ;
            ;3892   ; CHECK DATA FROM THE MEMORY
            ;3893   ;
            ;3894   ;****************************************
U 1171, 0800,003C,0180,0A28,0000,133D  ;3895  D_R[5]     ;CACHE OFF SBI ON
U 133D, 0000,003C,7580,3C00,0000,133E  ;3896  ID[MAINT]_D
U 133E, 0F10,0038,0180,0918,0200,133F  ;3897  VA_RC[03],D_0    ;READ LOW DATA FROM MEMORY
U 133F, 0000,003C,0180,C800,0000,1340  ;3898  MCT/READ.P
U 1340, 0001,003C,0180,09D8,0000,1084  ;3899  RC[0B]_D
U 1084, 0000,003D,0180,0800,0000,108C  ;3900 =00 CALL[NOINTR]
U 1085, 0000,003D,0180,0800,0000,1183  ;3901  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1086, 0000,003C,0180,0803,0000,1341  ;3902  VA_VA+4     ;READ UPPER DATA FROM MEMORY
U 1341, 0000,003C,0180,C800,0000,1342  ;3903 = MCT/READ.P
U 1342, 0001,003C,0180,09E0,0000,1088  ;3904  RC[0C]_D
U 1088, 0000,003D,0180,0800,0000,108C  ;3905 =00 CALL[NOINTR]
U 1089, 0000,003D,0180,0800,0000,1183  ;3906  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 108A, 0800,003C,0180,0A18,0000,1343  ;3907  D_R[3]     ;CHECK LOW DATA READ FROM MEMORY
U 1343, 0011,0020,0180,0958,0010,1344  ;3908 = ALU_D[XOR]RC[0B],CLK.UBCC
U 1344, 0800,013C,0180,0A20,0000,1172  ;3909  Z?,D_R[4]
U 1172, 0000,003D,0180,0800,0000,1183  ;3910 =0 ERROR1     ;LOW DATA READ FROM MEMORY BAD
U 1173, 0011,0020,0180,0960,0010,1345  ;3911  ALU_D[XOR]RC[0C],CLK.UBCC
U 1345, 0000,013C,0180,0800,0000,1174  ;3912  Z?
U 1174, 0000,003D,0180,0800,0000,1183  ;3913 =0 ERROR1     ;UPPER DATA READ FROM MEMORY BAD
            ;3914       ;==============================
            ;3915       ;DATA: EXPECTED UPPER DATA
            ;3916       ; EXPECTED LOW DATA
            ;3917       ; UPPER DATA READ
            ;3918       ; LOW DATA READ
            ;3919       ; D REG IMAGE USED FOR WRITE
            ;3920       ; BYTE DISPLACEMENT USED
            ;3921       ;==============================
U 1175, 0800,003C,0180,0A00,0000,1346  ;3922  D_R[0]     ;BYTE DISPLACEMENT = 7 ??
U 1346, 0019,0020,5D80,0800,0010,1347  ;3923  ALU_D.XOR.K[.7],CLK.UBCC
U 1347, 0000,013C,0180,0800,0000,1176  ;3924  Z?
U 1176, 0000,003C,0180,0800,0000,1348  ;3925 =0 J/C05200    ;NO, INCREMENT,REPEAT LOOP
U 1177, 0000,003C,0180,0800,0000,134C  ;3926  J/C05210    ;YES,GO CHECK DATA FLAG
U 1348, 0819,0014,0580,0A80,0000,1349  ;3927 C05200: R[0]_ALU,D_D+K[.1]  ;INCREMENT COUNT
U 1349, 0100,003C,0180,0800,0000,134A  ;3928  D_D.LEFT2
U 134A, 0500,003C,0180,0800,0000,134B  ;3929  D_D.LEFT
U 134B, 0001,003C,0180,0A88,0000,1159  ;3930  R[1]_D,J/C05020    ;GO REPEAT LOOP
U 134C, 0000,003C,0180,0A10,0010,134D  ;3931 C05210: ALU_R[2],CLK.UBCC  ;DATA FLAG SET ??
U 134D, 0000,013C,0180,0800,0000,1178  ;3932  Z?
U 1178, 0000,003C,0180,0800,0000,134E  ;3933 =0 J/C05END    ;ITS SET , END TEST
U 1179, 0018,0038,0580,0A90,0000,130A  ;3934  R[2]_K[.1],J/C05010  ;SET IT REPEAT TEST
U 134E, 0000,003C,0180,0800,0000,117A  ;3935 C05END: NOP    ; ALL DONE.
            ;3936
```

```
            ;3937  .PAGE "TEST 1D3 RESERVED"
            ;3938 =0
U 117A, 0000,003D,0180,0800,0000,1053  ;3939 T1D3: NEWTST
U 117B, 0000,003C,0180,0800,0000,117C  ;3940  NOP
            ;3941
```

```
            ;3942  .PAGE "TEST 1D4 RESERVED"
            ;3943 =0
U 117C, 0000,003D,0180,0800,0000,1053  ;3944 T1D4: NEWTST
U 117D, 0000,003C,0180,0800,0000,134F  ;3945  NOP
            ;3946
```

```
          ;3947  .PAGE "DUMMY NEWTST"
U 134F, 0000,003D,0180,0800,0000,1053  ;3948 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1524
ESKAR4D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:22           Page  10
:                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1907: 1874 1881= 1897= 2538= 2539= 2540= 1875
U 1008   2348= 2349= 2350= 2351= 2352= 2353= 2354= 2355=
U 1010   2545= 2546= 2547= 1876 1898= 1899= 1877 3649=
U 1018   2551= 2552= 2553= 1878 1951= 1953= 1879 3650=
U 1020   2557= 2560= 2562= 2563= 2751= 2752= 2753= 2755=
U 1028   2800= 2801= 2802= 2804= 3005= 3006= 3007= 3008=
U 1030   3014= 3015= 3016= 1880 3030= 3031= 3032= 1900
U 1038   3096= 3097= 3098= 1901 3258= 3259= 3260= 1902
U 1040   3264= 3265= 3266= 1903 3270= 3271= 3272= 1904
U 1048   3298= 3299= 3300= 1905 1964= 1965= 1972: 1906
U 1050   3326= 3327= 3328= 1940 3512= 3513= 3514= 1941
U 1058   1989= 1990= 1997: 1942 2013= 2014= 2020: 1943
U 1060   3522= 3523= 3524= 1944 3529= 3530= 3531= 1945
U 1068   3555= 3556= 3557= 1946 3602= 3603= 3604= 1947
U 1070   3799= 3800= 3801= 1948 3805= 3806= 3807= 1949
U 1078   3812= 3813= 3814= 1950 3820= 3821= 3822= 1954
U 1080   3848= 3849= 3857= 1955 3900= 3901= 3902= 1956
U 1088   3905= 3906= 3907= 1957 2102= 2103= 2106= 2107=
U 1090   2109= 2110= 2114= 2115= 2137= 2138= 2140= 2141=
U 1098   2143= 2144= 2154= 2155= 2156= 2158= 2159= 2160=
U 10A0   2161= 2162= 2163= 2164= 2165= 2166= 2167= 2168=
U 10A8   2169= 2170= 2287= 2288= 2327= 2328= 2336= 2337=
U 10B0   2338= 2340= 2343= 2344= 2385= 2386= 2387= 2388=
U 10B8   2395= 2396= 2402= 2404= 2407= 2408= 2427= 2428=
U 10C0   2430= 2431= 2597= 2598= 2601= 2602= 2613= 2614=
U 10C8   2617= 2618= 2664= 2665= 2706= 2707= 2956= 2957=
U 10D0   2958= 2959= 2985= 2986= 2989= 2990= 2993= 2994=
U 10D8   2998= 2999= 3000= 3001= 3003= 3004= 3010= 3011=
U 10E0   3019= 3020= 3023= 3028= 3035= 3036= 3040= 3045=
U 10E8   3053= 3054= 3058= 3065= 3071= 3072= 3078= 3079=
U 10F0   3083= 3089= 3101= 3102= 3105= 3111= 3113= 3114=
U 10F8   3211= 3212= 3213= 3214= 3231= 3232= 3234= 3235=
U 1100   1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108   1869: 1958 3248= 3249= 1870: 1871: 1872: 1873:
U 1110   3256= 3257= 3262= 3263= 3275= 3276= 3282= 3283=
U 1118   3285= 3286= 3303= 3304= 3313= 3314= 3331= 3332=
U 1120   3336= 3337= 3339= 3348= 3351= 3352= 3363= 3373=
U 1128   3376= 3377= 3472= 3473= 3474= 3475= 3494= 3495=
U 1130   3506= 3507= 3510= 3511= 3516= 3517= 3520= 3521=
U 1138   3534= 3535= 3542= 3543= 3560= 3561= 3565= 3566=
U 1140   3568= 3581= 3590= 3591= 3607= 3608= 3616= 3617=
U 1148   3630= 3631= 3635= 3637= 3639= 3647= 3658= 3659=
U 1150   3764= 3765= 3766= 3767= 1959 2040: 3780= 3781=
U 1158   3784= 3785= 3786= 3787= 3797= 3798= 3803= 3804=
U 1160   3817= 3818= 3825= 3826= 3832= 3833= 3843= 3844=
U 1168   3846= 3847= 3865= 3866= 3873= 3874= 3878= 3879=
U 1170   3881= 3895= 3910= 3911= 3913= 3922= 3925= 3926=
U 1178   3933= 3934= 3939= 3940= 3944= 3945= 1960 1961
U 1180   1962 1963 1968 1970 1971 1995 1996 2048
U 1188   2049 2056 2057 2063 2064 2065 2075 2076
U 1190   2077 2078 2079 2104 2105 2111 2112 2113
U 1198   2116 2117 2118 2119 2120 2121 2122 2123
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2135 2153 2171 2172 2201 2202 2203 2211
U 11A8  2212 2213 2214 2222 2223 2224 2225 2233
U 11B0  2234 2235 2236 2244 2245 2246 2247 2254
U 11B8  2255 2256 2257 2282 2329 2330 2331 2332
U 11C0  2341 2342 2346 2347 2360 2361 2363 2364
U 11C8  2365 2367 2372 2373 2374 2376 2381 2383
U 11D0  2384 2392 2393 2397 2398 2399 2400 2401
U 11D8  2409 2410 2411 2413 2414 2415 2416 2417
U 11E0  2418 2423 2425 2426 2429 2454 2455 2456
U 11E8  2457 2487 2488 2490 2491 2492 2536 2537
U 11F0  2564 2593 2595 2596 2600 2604 2605 2606
U 11F8  2607 2609 2610 2611 2612 2615 2616 2641
U 1200  2666 2667 2668 2669 2695 2696 2697 2698
U 1208  2699 2700 2701 2702 2704 2705 2831 2832
U 1210  2833 2852 2853 2854 2855 2960 2961 2962
U 1218  2963 2964 2965 2966 2967 2968 2969 2970
U 1220  2971 2972 2973 2974 2975 2976 2977 2978
U 1228  2979 2980 2984 2991 2992 2995 2996 2997
U 1230  3002 3009 3012 3013 3017 3018 3021 3022
U 1238  3029 3033 3034 3037 3038 3039 3046 3047
U 1240  3048 3049 3050 3051 3052 3055 3056 3057
U 1248  3067 3068 3069 3070 3073 3074 3075 3076
U 1250  3077 3080 3081 3082 3090 3091 3092 3093
U 1258  3094 3095 3099 3100 3103 3104 3112 3115
U 1260  3215 3216 3217 3218 3219 3220 3221 3222
U 1268  3223 3224 3228 3229 3230 3236 3237 3238
U 1270  3239 3240 3241 3242 3243 3244 3245 3246
U 1278  3250 3251 3252 3253 3254 3255 3261 3267
U 1280  3268 3269 3273 3274 3277 3278 3279 3280
U 1288  3281 3284 3287 3288 3301 3302 3305 3306
U 1290  3307 3308 3309 3310 3311 3312 3315 3316
U 1298  3329 3330 3333 3334 3335 3338 3349 3350
U 12A0  3361 3362 3374 3375 3378 3379 3380 3381
U 12A8  3383 3476 2041: 3477 3478 3479 3480 3481
U 12B0  3482 3483 3484 3485 3489 3490 3491 3492
U 12B8  3496 3497 3498 3499 3500 3501 3502 3503
U 12C0  3504 3505 3508 3509 3515 3518 3519 3525
U 12C8  3526 3527 3528 3532 3533 3536 3537 3538
U 12D0  3539 3540 3541 3545 3546 3558 3559 3562
U 12D8  3563 3564 3567 3582 3583 3584 3585 3586
U 12E0  3587 3588 3589 3592 3593 3605 3606 3609
U 12E8  3610 3611 3612 3613 3614 3615 3618 3619
U 12F0  3627 3628 3629 3632 3633 3634 3638 3648
U 12F8  3651 3652 3653 3654 3655 3656 3657 3660
U 1300  3768 3769 3770 3771 3772 3773 3774 3775
U 1308  3776 3777 3782 3783 3788 3789 3790 3791
U 1310  3792 3793 3794 3795 3796 3802 3808 3809
U 1318  3810 3815 3816 3819 3823 3824 3827 3828
U 1320  3829 3830 3834 3835 3836 3837 3838 3839
U 1328  3840 3841 3842 3845 3858 3859 3860 3861
U 1330  3862 3863 3864 3867 3868 3869 3870 3871
U 1338  3872 3875 3876 3877 3880 3896 3897 3898
U 1340  3899 3903 3904 3908 3909 3912 3923 3924
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3927 3928 3929 3930 3931 3932 3935 3948
U 1350 - 13EF Unused
U 13F0  2024: 2025: 2026: 2027: 2028: 2029: 2030: 2031:
U 13F8  2032: 2033: 2034: 2035: 2036: 2037: 2038: 2039:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR4D	864
Used	864
Remaining	

Total microwords used in memory U: 864
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                    SEQ 1528
ESKAR4D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:22   Page  10


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4D.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR4E.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 1529

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1917 Micro Monitor Interface Routines
; 1918   Newtest Routine
; 1975   Disable Cache Routine
; 1999   ERROR1 WITH PARAMETER
; 2025   ERROR 2 WITH PARAMETER
; 2098 DIAGNOSTIC SPECIFIC ROUTINES
; 2099   ENBFLT ROUTINE
; 2117   NOINTR ROUTINE
; 2163   Initialize the SBI
; 2191   RESET SUBTEST NUMBER
; 2214   Set Trap Mask
; 2243   Wait Loop Routine
; 2399   Set ECC Bits
; 2428   Set Diagnostic Mode Routine
; 2457   Select Controller
; 2495   SELECT LOWER CONTROLLER
; 2543   SELECT UPPER CONTROLLER
; 2591   FIND MS780-E MEMORY
; 2733   Generate IO Address
; 2759   Set Starting Address
; 2794   CONFIGURE MS780-E/H
; 2868   ENABLE NEXT MS780-E
; 2919 TEST 1D5 CACHE/SBI INVALIDATE TEST
; 3191 TEST 1D6 INTERRUPT DISABLE TESTS
; 3442 TEST 1D7 RESERVED
; 3447 TEST 1D8 RESERVED
; 3452 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR4E
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841  .PAGE
            ;1842
            ;1843  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7001  ;1862 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7001  ;1863 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7001  ;1864 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7001  ;1865 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7001  ;1866 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7001  ;1867 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7001  ;1868 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7001  ;1869 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7001  ;1870 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D. 0000.003C.8980.0800.0084.7001  ;1871 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7001  ;1872 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1016  ;1874 TRPH0: Q_ID[USTACK]
U 1016. 0C00.003C.01E0.0800.0000.101B  ;1875  Q_D,D_Q
U 101B. 0000.003C.8180.3C00.0000.101E  ;1876  ID[USTACK]_D
U 101E. 0C00.003C.01E0.0800.0000.1023  ;1877  Q_D,D_Q
U 1023. 0000.003C.01C0.0A78.0000.1027  ;1878  Q_R[0F]
U 1027. 0001.2024.0180.0800.0010.102F  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102F. 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1014  ;1897 TRPH1: RC[0D]_SC
U 1014, 0000.003D.D980.0800.0084.7138  ;1898 =0 SETRCF[.9]
U 1015, 0000.003C.49F0.2C00.0000.1033  ;1899  Q_ID[TBER0]
U 1033, 0001.203C.4DF0.2DB0.0000.1037  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1037, 0001.203C.65F0.2DB8.0000.103B  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 103B, 0001.203C.6DF0.2DD8.0000.103F  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 103F, 0001.203C.75F0.2DE0.0000.1043  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1043, 0001.203C.79F0.2DC8.0000.1047  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1047, 0001.203C.69F0.2DC0.0000.104B  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 104B, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1130  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;+
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
```

```
            ;1916  .PAGE
            ;1917  .TOC " Micro Monitor Interface Routines"
            ;1918  .TOC "   Newtest Routine"
            ;1919 ;++
            ;1920 ; FUNCTIONAL DESCRIPTION:
            ;1921 ;
            ;1922 ; This routine Initializes the following registers:
            ;1923 ;  PSL   to 1F
            ;1924 ;  CES   to 0
            ;1925 ;  ACC.CS   to disable accellerator
            ;1926 ;  SIR   to 0
            ;1927 ;  CLK.CS   to stop interval timer
            ;1928 ;  TBER0   to disable the TB
            ;1929 ;  SBI.MAINT to    0
            ;1930 ;  SBI.ERR   to 0
            ;1931 ;  FAULT   to 0
            ;1932 ;  COMP   to 0
            ;1933 ;  RC[0E]   to  0
            ;1934 ;  R[0F]   to 0
            ;1935 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1936 ; A SCOPE call is then made to the Monitor.
            ;1937 ;
            ;1938 ; CALLING CONSTRAINT:
            ;1939 ;
            ;1940 ; =0
            ;1941 ;
            ;1942 ; SIDE EFFECTS:
            ;1943 ;
            ;1944 ; D Reg is destroyed
            ;1945 ; SC is destroyed
            ;1946 ;--
U 104F, 0000.003C.65F8.0800.0084.705B  ;1947 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 105B, 0818.0038.8D80.0800.0000.105F  ;1948  d_k[.1f]  ; D=1F
U 105F, 0D00.003C.0180.0800.0000.1109  ;1949  d_dal.sc  ; D=001F0000
U 1109, 0000.003C.3D80.3C00.0000.111A  ;1950  id[psl]_d  ; psl = 001F0000
U 111A, 0818.0038.0580.0800.0000.111B  ;1951  d_k[.1]  ; Clear the CES register
U 111B, 0D00.003C.0180.0800.0000.111C  ;1952  d_dal.sc  ; D = 00010000
U 111C, 0F00.003C.3180.3C00.0000.111D  ;1953  id[ces]_d,d_0  ; ces = 00010000
U 111D, 0000.003C.5D80.3C00.0000.111E  ;1954  id[acc.cs]_d  ; acc.cs = 0
U 111E, 0000.003C.3980.3C00.0000.111F  ;1955  id[sir]_d  ; sir = 0
U 111F, 0000.003C.2980.3C00.0000.1120  ;1956  id[clk.cs]_d  ; clk.cs = 0
U 1120, 0000.003C.4980.3C00.0000.101C  ;1957  id[tber0]_d  ; tber0 = 0
U 101C, 0000.003D.0180.0800.0000.1158  ;1958 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1959  intrpt.strobe,  ; Strobe interrupts
U 101D, 0818.0038.C180.0800.4000.1121  ;1960      d_k[.ffff]  ; Setup to clear SBI error register and
U 1121, 0801.0028.6580.3C00.0000.1122  ;1961  id[sbi.err]_d,d_not.d ; and fault register
U 1122, 0F00.003C.6D80.3C00.0000.1123  ;1962  id[fault]_d,d_0  ; ...
U 1123, 0000.003C.6D80.3C00.0000.1124  ;1963  id[fault]_d  ; fault = 0
U 1124, 0000.003C.7580.3C00.0000.1125  ;1964  id[maint]_d  ; MAINT = 0
U 1125, 0000.003C.6580.3C00.0000.1126  ;1965  id[sbi.err]_d  ; sbi error reg = 0
U 1126, 0000.003C.7180.3C00.0000.1127  ;1966  id[comp]_d  ; comp reg = 0
U 1127, 0000.003C.E580.3C00.0000.1128  ;1967  id[T9]_d  ; Clear code for subroutine START.TEST
U 1128, 0001.003C.0180.09F0.0000.1129  ;1968  rc[0e]_d  ;
U 1129, 0001.003C.0180.0AF8.0000.112A  ;1969  r[0f]_d  ; Clear expected trap mask
U 112A, 0001.003C.0180.09F8.0000.104C  ;1970  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 104C, 0000,003D,0180,0800,0000,112B  ;1971 =0 call[disable.cache] ; Disable the cache
U 104D, 0F03,003C,0180,09E8,0000,105E  ;1972  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
           ;1973
```

```
            ;1974  .PAGE
            ;1975  .TOC "   Disable Cache Routine"
            ;1976 ;++
            ;1977 ; FUNCTIONAL DESCRIPTION:
            ;1978 ;
            ;1979 ; This routine disables cache hits by setting bits <16:15>
            ;1980 ; in the maintenance register.
            ;1981 ;
            ;1982 ; CALLING SEQUENCE:
            ;1983 ;
            ;1984 ; =0
            ;1985 ;
            ;1986 ; SIDE EFFECTS:
            ;1987 ;
            ;1988 ; D & Q Registers destroyed
            ;1989 ;--
            ;1990 DISABLE.CACHE:
U 112B, 0818,0038,4580,0800,0000,112C  ;1991  d_k[.8000]  ; ... and seed constant
U 112C, 0500,003C,0180,0800,0000,112D  ;1992  d_d.left  ; Generate final constant
U 112D, 0819,0030,4580,0800,0000,112E  ;1993  d_d.or.k[.8000]  ; ... = 00018000
U 112E, 0000,003E,7580,3C00,0000,0001  ;1994  id[maint]_d,return1 ; Disable cache and return
            ;1995
U 112F, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
            ;1997
```

```
            ;1998  .PAGE
            ;1999  .TOC "   ERROR1 WITH PARAMETER"
            ;2000 ;++
            ;2001 ; FUNCTIONAL DESCRIPTION:
            ;2002 ;
            ;2003 ; This subroutine takes  as parameter the number of arguments
            ;2004 ; to be printed to the console in the case of an error logout.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; =0
            ;2009 ;INPUTS:
            ;2010 ;
            ;2011 ; RC[OF] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2012 ;--
            ;2013 =0
            ;2014 ERR1.ARG:
U 1058, 0000,003D,0180,0800,0000,1138  ;2015  CALL[SETRCF]  ; Set the number of arguments in RC[OF]
U 1059, 0000,003C,0180,0800,0000,1130  ;2016  J/ERROR1  ; Go to ERROR1
            ;2017 ;
            ;2018 ;
            ;2019
U 1130, 0810,0038,0180,0978,0000,1131  ;2020 ERROR1: D_RC[OF]
U 1131, 0819,0034,4D80,0800,0000,104E  ;2021  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2022 104E: D_D.OR.K[.4],J/CALLCON
            ;2023
```

```
            ;2024  .PAGE
            ;2025  .TOC "   ERROR 2 WITH PARAMETER"
            ;2026 ;**
            ;2027 ; FUNCTIONAL DESCRIPTION:
            ;2028 ;
            ;2029 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2030 ; except that in this case after setting the number of arguments
            ;2031 ; too the console, control is transferred to routine ERROR2.
            ;2032 ;
            ;2033 ; INPUTS:
            ;2034 ;
            ;2035 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2036 ;
            ;2037 ;--
            ;2038 =0
            ;2039 ERR2.ARG:
U 105C. 0000.003D.0180.0800.0000.1138  ;2040  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 105D. 0000.003C.0180.0800.0000.1132  ;2041  J/ERROR2  ; Go to ERROR2
            ;2042 ;
            ;2043 ;
            ;2044
U 1132. 0810.0038.0180.0978.0000.1133  ;2045 ERROR2: D_RC[0F]
U 1133. 0819.0034.4D80.0800.0000.105A  ;2046  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;2047 105A: D_D.OR.K[.6],J/CALLCON
            ;2048 105E:
            ;2049 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;2050  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2051 ;*
            ;2052 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2053 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1  ;2054 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;2055 13F1: NOP
U 13F2. 0000.003C.0180.0800.0000.13F3  ;2056 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;2057 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;2058 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;2059 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;2060 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;2061 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;2062 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;2063 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;2064 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;2065 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;2066 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;2067 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;2068 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1155  ;2069 13FF: NOP
U 1155. 0000.003C.0180.0800.0000.12AA  ;2070 1155: NOP
U 12AA. 0000.003C.0180.0800.0000.1134  ;2071 12AA: NOP
            ;2072 ;*
            ;2073 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2074 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2075 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2076 ; TYPING IT.
            ;2077 ;-
U 1134. 0810.0038.4D80.0978.0000.1135  ;2078 SUBTST: D_RC[0F],KMX/.FF00
```

```
U 1135, 0019,4016,4D80,09F8,0000,0001  ;2079  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2080 ;*****************************************************************
            ;2081 ;+       THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2082 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2083 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2084 ;-
            ;2085 SUBTEST1:
U 1136, 0810,0038,0180,0978,0000,1137  ;2086  D_RC[0F]
U 1137, 0019,0032,4D80,09F8,0000,0001  ;2087  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2088 ;+
            ;2089 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2090 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2091 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2092 ;-
U 1138, 0010,0038,01C0,0978,0000,1139  ;2093 SETRCF: Q_RC[0F]
U 1139, 0019,2034,4DC0,0800,0000,113A  ;2094  Q_Q.AND.K[.FF00]
U 113A, 0019,2032,1D80,09F8,0000,0001  ;2095  RC[0F]_Q.OR.SC,RETURN1
            ;2096
```

```
            ;2097  .PAGE
            ;2098  .TOC " DIAGNOSTIC SPECIFIC ROUTINES"
            ;2099  .TOC "    ENBFLT ROUTINE"
            ;2100 ;*****************************************************************
            ;2101 ;+
            ;2102 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2103 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2104 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2105 ;
            ;2106 ; D AND SC GET CLOBBERED
            ;2107 ;-
            ;2108 ;*****************************************************************
U 113B, 0F00,003C,2180,0800,0084,713C  ;2109 ENBFLT: SC_K[.14],D_0
U 113C, 0000,003C,0580,0800,0084,B13D  ;2110  SC_SC-K[.1]
U 113D, 0801,001C,0180,0800,0000,113E  ;2111  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 113E, 0600,003C,6D80,3C00,0000,113F  ;2112  ID[FAULT]_D,D_D.RIGHT
U 113F, 0000,003E,6D80,3C00,0000,0001  ;2113  ID[FAULT]_D,RETURN1
            ;2114
            ;2115
```

```
            ;2116  .PAGE
            ;2117  .TOC "    NOINTR ROUTINE"
            ;2118 ;*****************************************************************
            ;2119 ;*
            ;2120 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
            ;2121 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
            ;2122 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
            ;2123 ; IF ANY INTERRUPTS OCCURRED:
            ;2124 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
            ;2125 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
            ;2126 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
            ;2127 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
            ;2128 ;  RETURN1 IS USED TO GET BACK
            ;2129 ; ELSE
            ;2130 ;  RETURN2 IS USED
            ;2131 ;
            ;2132 ; THE D AND Q REGISTERS GET CLOBBERED.
            ;2133 ;
            ;2134 ;-
            ;2135 ;*****************************************************************
            ;2136 =0
U 1060. 0000,003D,0180,0800,0000,1145  ;2137 NOINTR: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 1061. 0000,003C,35F0,2C00,0000,1140  ;2138  Q_ID[VECTOR]
U 1140. 0019,2034,8180,0800,0010,1141  ;2139  ALU_Q.AND.K[.3FF],CLK.UBCC
U 1141. 0000,013C,0180,0800,0000,1062  ;2140  Z?
U 1062. 0000,003C,0180,0800,0000,1149  ;2141 =0 J/ERRFLT    ; INTR. OCCURRED,SET FLAG,GET DATA
U 1063. 0000,003E,0180,0800,0000,0002  ;2142  RETURN2     ; NO INTR, RETURN2
            ;2143 =0
U 1064. 0000,003D,0180,0800,0000,1145  ;2144 WRTOUT: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 1065. 0000,003C,35F0,2C00,0000,1142  ;2145  Q_ID[VECTOR]
U 1142. 0819,2034,8180,0800,0000,1143  ;2146  ALU_Q.AND.K[.3FF],D_ALU
U 1143. 0019,0020,A580,0800,0010,1144  ;2147  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 1144. 0000,013C,0180,0800,0000,1066  ;2148  Z?
U 1066. 0000,003C,0180,0800,0000,1149  ;2149 =0 J/ERRFLT    ; WRONG VECTOR,SET FLAG,GET DATA
U 1067. 0000,003E,0180,0800,0000,0002  ;2150  RETURN2     ; OK, RETURN2
U 1145. 0F00,003C,0180,0800,0000,1146  ;2151 STROBE: D_0     ; SET LOWEST IPL ACTIVE
U 1146. 0000,003C,3D80,3C00,0000,1147  ;2152  ID[PSL]_D    ;
U 1147. 0000,003C,0180,0800,4000,1148  ;2153  IEK/ISTR    ; STROBE INTERRUPT
U 1148. 0000,003E,0180,0800,0000,0001  ;2154  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 1149. 0018,0038,0580,09A8,0000,114A  ;2155 ERRFLT: RC[5]_K[.1]    ; Set Flag
U 114A. 0001,203C,65F0,2DC0,0000,114B  ;2156  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 114B. 0001,203C,6DF0,2DB8,0000,114C  ;2157  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 114C. 0001,203E,0180,09B0,0000,0001  ;2158  RC[6]_Q,RETURN1    ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
            ;2159
U 114D. 0818,0038,0180,0800,0000,105E  ;2160 DELAY: D_K[.8],J/CALLCON
            ;2161
```

```
            ;2162  .PAGE
            ;2163  .TOC "   Initialize the SBI"
            ;2164 ;++
            ;2165 ; FUNCTIONAL DESCRIPTION:
            ;2166 ;
            ;2167 ; This routine performs the following functions:
            ;2168 ;
            ;2169 ;  Executes an SBI Unjam
            ;2170 ;  Clears the SBI Fault Latch
            ;2171 ;  Clears the SBI Error Register
            ;2172 ;
            ;2173 ; CALLING CONSTRAINT:
            ;2174 ;
            ;2175 ; =0
            ;2176 ;
            ;2177 ; SIDE EFFECTS:
            ;2178 ;
            ;2179 ; D & Q Register destroyed
            ;2180 ;--
            ;2181 =0
            ;2182 SBI.INIT:
U 1068, 0000,003D,0180,0800,0000,1158  ;2183  call[sbi.unjam]  ; Unjam the SBI
U 1069, 0818,0038,C180,0800,0000,114E  ;2184  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 114E, 0801,0028,6580,3C00,0000,114F  ;2185  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 114F, 0F00,003C,6D80,3C00,0000,1150  ;2186  id[fault]_d,d_0
U 1150, 0000,003C,6D80,3C00,0000,1151  ;2187  id[fault]_d
U 1151, 0000,003E,6580,3C00,0000,0001  ;2188  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2189
```

```
            ;2190  .PAGE
            ;2191  .TOC "   RESET SUBTEST NUMBER"
            ;2192 ;**
            ;2193 ; FUNCTIONAL DESCRIPTION:
            ;2194 ;
            ;2195 ; Since some  of the tests will have to be restarted when two
            ;2196 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2197 ; the subtest counter to zero ( only for thoes tests that have
            ;2198 ; more than one subtest). This subroutine may also be necessary
            ;2199 ; when a test is being loop on.
            ;2200 ;
            ;2201 ; CALLING CONSTRAINT:
            ;2202 ;
            ;2203 ; =0
            ;2204 ; SIDE EFFECTS:
            ;2205 ;
            ;2206 ; D is destroyed
            ;2207 ;
            ;2208 ;--
            ;2209 RESET.SUBTST:
            ;2210  RC[0F]_0,   ; Reset subtest number
U 1152. 0003,003E,0180,09F8,0000,0001  ;2211   RETURN1   ; ...and return
            ;2212
```

```
            ;2213  .PAGE
            ;2214  .TOC "   Set Trap Mask"
            ;2215 ;++
            ;2216 ; FUNCTIONAL DESCRIPTION:
            ;2217 ;
            ;2218 ; This routine is used to set a bit in the Micro Trap Mask
            ;2219 ; R[0F].
            ;2220 ;
            ;2221 ; CALLING CONSTRAINT:
            ;2222 ;
            ;2223 ; =0
            ;2224 ;
            ;2225 ; INPUTS:
            ;2226 ;
            ;2227 ; SC - Contains bit number to set.
            ;2228 ;
            ;2229 ; OUTPUTS:
            ;2230 ;
            ;2231 ; rc[0E] - Contains the current trap mask
            ;2232 ;
            ;2233 ; SIDE EFFECTS:
            ;2234 ;
            ;2235 ; d regsiter is destroyed
            ;2236 ;--
            ;2237 SET_TRAP_MASK:
U 1153, 0800.003C.0180.0A78.0000.1154  ;2238  d_r[0f]
U 1154, 0801.001C.0180.0AF8.0000.1156  ;2239  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1156, 0001.003E.0180.0AF0.0000.0001  ;2240  r[0E]_d,return1    ; Save mask and return
            ;2241
```

```
            ;2242  .PAGE
            ;2243  .TOC "   Wait Loop Routine"
            ;2244 ;**
            ;2245 ; FUNCTIONAL DESCRIPTION:
            ;2246 ;
            ;2247 ; This routine is used to wait the specified number of machine cycles.
            ;2248 ; This routine is typically called to wait for an SBI write to
            ;2249 ; I/O space to complete.
            ;2250 ;
            ;2251 ; CALLING CONSTRAINT:
            ;2252 ;
            ;2253 ; =0
            ;2254 ;
            ;2255 ; INPUTS:
            ;2256 ;
            ;2257 ; d - Contains number of cycles to wait
            ;2258 ; alu.z condition code must not be set
            ;2259 ;
            ;2260 ; SIDE EFFECTS:
            ;2261 ;
            ;2262 ; d is destroyed
            ;2263 ;--
            ;2264 WAIT.LOOP:
U 1157, 0018,0038,6180,0800,0010,106A  ;2265  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2266      ; ...is not set
            ;2267 =0
            ;2268 W.LOOP:
            ;2269  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 106A, 0819,0100,0580,0800,0010,106A  ;2270       j/W.LOOP
U 106B, 0000,003E,0180,0800,0000,0001  ;2271  return1    ; exit
            ;2272
            ;2273
            ;2274 SBI.UNJAM:
            ;2275 CLRSBI:
            ;2276 ;============================================================================
            ;2277 ;////////////////////////////////////////////////////////////////////////////
            ;2278 ;*
            ;2279 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2280 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2281 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2282 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2283 ;-
U 1158, 0818,0038,6580,8000,0010,106C  ;2284 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2285 =0
U 106C, 0819,0100,0580,8000,0010,106C  ;2286 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 106D, 0818,0038,6580,8800,0010,106E  ;2287  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2288 =0
U 106E, 0819,0100,0580,8800,0010,106E  ;2289 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 106F, 0818,0039,6580,8000,0010,1070  ;2290  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2291 =0
U 1070, 0819,0100,0580,8000,0010,1070  ;2292 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 1071, 0000,003E,0180,0800,0000,0001  ;2293  RETURN1    ; DONE.
            ;2294 INIT: ;=================================================================
            ;2295  ;
            ;2296  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
```

```
            ;2297   ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2298   ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2299   ;
            ;2300   ;=============================================================================
            ;2301   ;
U 1159, 0003,003C,0180,0AF8,0000,1072   ;2302   R[0F]_0     ;CLEAR U-TRAPS
U 1072, 0000,003D,0180,0800,0000,1158   ;2303  =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 1073, 0000,003C,0180,0800,0000,1074   ;2304   NOP
U 1074, 0000,003D,0180,0800,0000,115C   ;2305  =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2306   ;   INTRUPT REG
U 1075, 0000,003C,0180,0800,0000,1076   ;2307   NOP
U 1076, 0000,003D,0180,0800,0000,1163   ;2308  =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 1077, 0000,003C,0180,0800,0000,1078   ;2309   NOP
U 1078, 0000,003D,0180,0800,0000,1167   ;2310  =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 1079, 0000,003C,0180,0800,0000,107A   ;2311   NOP
U 107A, 0000,003D,0180,0800,0000,115F   ;2312  =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 107B, 0000,003C,0180,0800,0000,107C   ;2313   NOP
U 107C, 0000,003D,0180,0800,0000,116B   ;2314  =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 107D, 0000,003C,0180,0800,0000,107E   ;2315   NOP
U 107E, 0000,003D,0180,0800,0000,113B   ;2316  =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 107F, 0818,0038,4580,0800,0000,1080   ;2317   D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;2318  =0 CALL[DISABLE.CACHE],   ;SHUT CACHE OFF
U 1080, 0000,003D,7980,3C00,0000,112B   ;2319    ID[PARITY]_D
U 1081, 0F00,003C,0180,0800,0000,115A   ;2320   D_0
U 115A, 0000,003C,4980,3C00,0000,115B   ;2321   ID[TBER0]_D    ;SHUT TB OFF
U 115B, 0000,003E,7180,3C00,0000,0001   ;2322   ID[COMP]_D,RETURN1   ;DISABLE SILO INTERRUPTS
            ;2323   ;
            ;2324   ;=============================================================================
            ;2325   ;*
            ;2326   ;CHECK FOR INTERRUPTS PENDING
            ;2327   ;
            ;2328   ; INTERRUPT  VECTOR CHECKED
            ;2329   ; ------------------------------------
            ;2330   ; WRITE TIMEOUT  ^X60
            ;2331   ; SBI FAULT  ^X5C
            ;2332   ; CRD,RDS TAG  ^X54
            ;2333   ; SILO    ^X50
            ;2334   ;
            ;2335   ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2336   ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2337   ;
            ;2338   ; DATA: EXPECTED VECTOR
            ;2339   ;  VECTOR STROBED
            ;2340   ;-
            ;2341   ;=============================================================================
            ;2342   ;
            ;2343  SETPSL: ;=============================================================================
            ;2344   ;
            ;2345   ;DROP THE CPU IPR LEVEL TO
            ;2346   ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2347   ; PENDING
            ;2348   ;
            ;2349   ;=============================================================================
            ;2350   ;
U 115C, 0F00,003C,0180,0800,0000,115D   ;2351   D_0
```

```
U 115D, 0000,003C,3980,3C00,0000,115E  ;2352  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 115E, 0000,003E,3D80,3C00,0000,0001  ;2353  RETURN1,ID[PSL]_D
            ;2354 CLRTIM: ;=================================================================
            ;2355  ;
            ;2356  ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2357  ; SBI ERROR REGISTER
            ;2358  ;
            ;2359  ;=================================================================
            ;2360  ;
U 115F, 0818,0038,0580,0800,0000,1160  ;2361  D_K[.1]
U 1160, 0000,003C,85F8,0800,0084,7161  ;2362  Q_0,SC_K[.C]
U 1161, 0D00,003C,0180,0800,0000,1162  ;2363  D_DAL.SC
U 1162, 0000,003E,6580,3C00,0000,0001  ;2364  ID[SBI.ERR]_D,RETURN1
            ;2365 CLRFLT: ;=================================================================
            ;2366  ;
            ;2367  ;CLEAR THE CP FAULT BIT IN THE
            ;2368  ; FAULT REGISTER
            ;2369  ;
            ;2370  ;=================================================================
            ;2371  ;
U 1163, 0818,0038,0180,0800,0000,1164  ;2372  D_K[.8]
U 1164, 0000,003C,65F8,0800,0084,7165  ;2373  Q_0,SC_K[.10]
U 1165, 0D00,003C,0180,0800,0000,1166  ;2374  D_DAL.SC
U 1166, 0000,003E,6D80,3C00,0000,0001  ;2375  ID[FAULT]_D,RETURN1
            ;2376 CLRECC: ;=================================================================
            ;2377  ;
            ;2378  ;CLEAR CRD,RDS BIT IN THE
            ;2379  ;SBI ERROR REGISTER
            ;2380  ;
            ;2381  ;=================================================================
            ;2382  ;
U 1167, 0818,0038,0D80,0800,0000,1168  ;2383  D_K[.3]
U 1168, 0000,003C,89F8,0800,0084,7169  ;2384  Q_0,SC_K[.D]
U 1169, 0D00,003C,0180,0800,0000,116A  ;2385  D_DAL.SC
U 116A, 0000,003E,6580,3C00,0000,0001  ;2386  ID[SBI.ERR]_D,RETURN1
            ;2387 ENBECC: ;=================================================================
            ;2388  ;
            ;2389  ;ENABLE CRD,RDS INTERRUPTS
            ;2390  ;
            ;2391  ;=================================================================
            ;2392  ;
U 116B, 0818,0038,0580,0800,0000,116C  ;2393  D_K[.1]
U 116C, 0000,003C,61F8,0800,0084,716D  ;2394  Q_0,SC_K[.F]
U 116D, 0D00,003C,0180,0800,0000,116E  ;2395  D_DAL.SC
U 116E, 0000,003E,6580,3C00,0000,0001  ;2396  ID[SBI.ERR]_D,RETURN1
            ;2397
```

```
            ;2398  .PAGE
            ;2399  .TOC "   Set ECC Bits"
            ;2400 ;++
            ;2401 ; FUNCTIONAL DESCRIPTION:
            ;2402 ;
            ;2403 ; This routine is used to set the specified ECC bits
            ;2404 ; in Register B. Other bits in the register remain unchanged.
            ;2405 ;
            ;2406 ; CALLING CONSTRAINT:
            ;2407 ;
            ;2408 ; =0
            ;2409 ;
            ;2410 ; INPUTS:
            ;2411 ;
            ;2412 ; d - Contains the bits to set in <6:0>
            ;2413 ; rc[0] - Contains base address of controller
            ;2414 ;
            ;2415 ; SIDE EFFECTS:
            ;2416 ;
            ;2417 ; d,va,sc are destroyed
            ;2418 ;--
            ;2419 SET_ECC:
U 116F, 0010,0038,0180,0970,0200,1170  ;2420  va_rc[0E]  ; Set address of Register B
U 1170, 0000,003C,0180,0803,0000,1171  ;2421  va_va+4    ; ...in VA
U 1171, 0000,003C,01E0,C800,0000,1172  ;2422  q_d.d[long]_cache.p ; Read current contents of register
U 1172, 0819,0024,5980,0800,0000,1173  ;2423  d_d.andnot.k[.7f] ; Clear current ECC bits
U 1173, 081D,0030,0180,0800,0000,1174  ;2424  d_d.or.q  ; Insert new ecc bits
U 1174, 0000,003E,0180,A800,0000,0001  ;2425  cache.p_d[long],return1 ; Set the specified bits and return
            ;2426
```

```
            ;2427  .PAGE
            ;2428  .TOC "   Set Diagnostic Mode Routine"
            ;2429 ;++
            ;2430 ; FUNCTIONAL DESCRIPTION:
            ;2431 ;
            ;2432 ; This routine is used to set the specified diagnostic mode
            ;2433 ; in Register B. Other bits in the register remain unchanged.
            ;2434 ;
            ;2435 ; CALLING CONSTRAINT:
            ;2436 ;
            ;2437 ; =0
            ;2438 ;
            ;2439 ; INPUTS:
            ;2440 ;
            ;2441 ; d - Contains the mode to set in bits<1:0>
            ;2442 ; rc[0e] - Contains base address of controller
            ;2443 ;
            ;2444 ; SIDE EFFECTS:
            ;2445 ;
            ;2446 ; d,va,sc are destroyed
            ;2447 ;--
            ;2448 SET_DIAG_MODE:
U 1175,  0010.0038.D9F8.0970.0284.7176  ;2449  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 1176,  0D00.003C.0180.0803.0000.1177  ;2450  va_va+4,d_dal.sc ; Shift specified mode into position
U 1177,  0000.003C.01E0.0800.0000.1178  ;2451  q_d   ; Save the diagnostic mode bits
U 1178,  0000.003C.0180.C800.0000.1179  ;2452  d[long]_cache.p  ; Read the contents of the CNFG register B
U 1179,  081D.0030.0180.0800.0000.117A  ;2453  d_d.or.q  ; Set the diagnostic mode bits
U 117A,  0000.003E.0180.A800.0000.0001  ;2454  cache.p_d[long],return1 ; Set the specified mode and return
            ;2455
```

```
            ;2456  .PAGE
            ;2457  .TOC "   Select Controller"
            ;2458 ;**
            ;2459 ; FUNCTIONAL DESCRIPTION:
            ;2460 ;
            ;2461 ; This routine is used to put the memory system into the
            ;2462 ; specified configuration with respect to controller select.
            ;2463 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2464 ; a RETURN2 is made.
            ;2465 ;
            ;2466 ; CALLING CONSTRAINT:
            ;2467 ;
            ;2468 ; =00
            ;2469 ;
            ;2470 ; INPUTS:
            ;2471 ;
            ;2472 ; d - Contains value to write to bits<2:0> of Register A
            ;2473 ; rc[0e] - Address of Register A
            ;2474 ;
            ;2475 ; SIDE EFFECTS:
            ;2476 ;
            ;2477 ; sc,d,va are destroyed
            ;2478 ;--
            ;2479 SELECT.CNTRLR:
U 117B, 0010,0038,0180,0970,0284,717C  ;2480  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 117C, 0801,001C,0180,0800,0000,117D  ;2481  d_d.ornot.mask  ; Insert the enable bit into D reg
U 117D, 0000,003C,0180,A800,0000,117E  ;2482  cache.p_d[long]  ; Write the configuration Register
U 117E, 0818,0038,5D80,0800,0000,117F  ;2483  d_k[.7]  ; Get Misconfiguration bits
U 117F, 0000,003C,61F8,0800,0084,7180  ;2484  sc_k[.F],q_0  ; ...
U 1180, 0D00,003C,0180,0800,0000,1181  ;2485  d_dal.sc  ; ... D = x38000
U 1181, 0000,003C,01E0,0800,0000,1182  ;2486  q_d  ; Save mask
U 1182, 0000,003C,0180,C800,0000,1183  ;2487  d[long]_cache.p  ; Read CNFG A Reg and
            ;2488  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1183, 001D,2034,0180,0800,0010,1184  ;2489  clk.ubcc ; ...
U 1184, 0000,013C,0180,0800,0000,1082  ;2490  z?  ; Branch if no
U 1082, 0000,003E,0180,0800,0000,0001  ;2491 =0 RETURN1  ; Bad configuration
U 1083, 0000,003E,0180,0800,0000,0002  ;2492  RETURN2  ; Configuration ok
            ;2493
```

```
;2494  .PAGE
;2495  :TOC "   SELECT LOWER CONTROLLER"
;2496 ;*************************************************************
;2497 ;++
;2498 ;
;2499 ; Functional Description:
;2500 ; ----------------------
;2501 ;
;2502 ; This subroutine can be called to select the lower
;2503 ; Controller on a MS780-E/H.
;2504 ; If the Lower controller is misconfigured then a
;2505 ; RETURN1 will be made.  Otherwise a RETURN2
;2506 ;
;2507 ; Calling Constraint: =00
;2508 ; ------------------
;2509 ;
;2510 ;
;2511 ; Inputs:
;2512 ; ------
;2513 ;
;2514 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2515 ;
;2516 ; Outputs:
;2517 ; -------
;2518 ;
;2519 ; RC[0D] will have the address of CNFG Register C
;2520 ;        ( 2000x008 )
;2521 ;
;2522 ; Side Effects:
;2523 ; ------------
;2524 ;
;2525 ; Contents of the following registers will lost:
;2526 ;
;2527 ; D
;2528 ;
;2529 ; The Lower controller on the MS780-E/H will be configured
;2530 ; when the subroutine is exitted with a RETURN2
;2531 ;--
;2532 ;*************************************************************
;2533 =00
;2534 SELECT.LOWER:
;2535   D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,117B  ;2536   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2537  RETURN1     ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2538  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2539   RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2540   RETURN2    ; Let caller know that the controller
;2541        ; ...was configured properly
```

```
;2542  .PAGE
;2543  .TOC "   SELECT UPPER CONTROLLER"
;2544 ;*************************************************************
;2545 ;++
;2546 ;
;2547 ; Functional Description:
;2548 ; ----------------------
;2549 ;
;2550 ; This subroutine can be called to select the upper
;2551 ; Controller on a MS780-E/H.
;2552 ; If the Upper controller is misconfigured then a
;2553 ; RETURN1 will be made.  Otherwise a RETURN2
;2554 ;
;2555 ; Calling Constraint: =00
;2556 ; ------------------
;2557 ;
;2558 ;
;2559 ; Inputs:
;2560 ; ------
;2561 ;
;2562 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2563 ;
;2564 ; Outputs:
;2565 ; -------
;2566 ;
;2567 ; RC[0D] will have the address of CNFG Register D
;2568 ;        ( 2000x010 )
;2569 ;
;2570 ; Side Effects:
;2571 ; ------------
;2572 ;
;2573 ; Contents of the following registers will lost:
;2574 ;
;2575 ; D
;2576 ;
;2577 ; The Upper controller on the MS780-E/H will be configured
;2578 ; when the subroutine is exitted with a RETURN2
;2579 ;--
;2580 ;*************************************************************
;2581 =00
;2582 SELECT.UPPER:
;2583  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,117B  ;2584   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2585  RETURN1     ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2586  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2587  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2588   RETURN2    ; Let caller know that the controller
;2589       ; ...was configured properly
```

```
            ;2590  .PAGE
            ;2591  .TOC "   FIND MS780-E MEMORY"
            ;2592 ;++
            ;2593 ; FUNCTIONAL DESCRIPTION:
            ;2594 ;
            ;2595 ;          The diagnostic is designed to handle up to 4 (four)
            ;2596 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2597 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2598 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2599 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2600 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2601 ;     first MS780-E found will have its starting address set to 0
            ;2602 ;     (zero).
            ;2603 ;     All the other MS780-E/H's found will have their starting address
            ;2604 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2605 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2606 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2607 ;     current MS780-E unit under test.
            ;2608 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2609 ;     set to maximum.
            ;2610 ;
            ;2611 ; CALLING CONSTRAINT:
            ;2612 ;
            ;2613 ; =00
            ;2614 ;
            ;2615 ; OUTPUTS:
            ;2616 ;
            ;2617 ; rc[0e] - Contains address of Configuration Register
            ;2618 ; r[0]  - Contains TR level
            ;2619 ;
            ;2620 ; SIDE EFFECTS:
            ;2621 ;
            ;2622 ; D register is destroyed.
            ;2623 ;--
            ;2624 =0
            ;2625 FIND.MS780E:
U 1084,  0000,003D,0180,0800,0000,1068  ;2626  call[sbi.init]  ; Make sure SBI is enabled
U 1085,  0003,003C,0180,0988,0000,1185  ;2627  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1185,  0818,0038,1980,0800,0000,1186  ;2628  d_k[zero]  ; Clear MS780-E/H counter
U 1186,  0000,003C,F980,3C00,0000,1187  ;2629  id[3e]_d  ; ...
U 1187,  0018,0038,0580,0A80,0000,1188  ;2630  r[0]_k[.1]  ; Init TR counter
U 1188,  0F03,003C,0180,09F0,0000,1086  ;2631  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2632      ; ...CNFG A Reg
            ;2633 =0
            ;2634 FIND.MEM.LOOP:
U 1086,  0800,003D,0180,0A00,0000,11AE  ;2635  d_r[0],call[generate.io]; Generate address of TR level
U 1087,  0001,003C,0180,09F0,0200,1088  ;2636  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1088,  0000,003D,8980,0800,0084,7153  ;2637 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2638  d[long]_cache.p, ; Read the specified tr level
U 1089,  0000,003C,0180,C970,0000,1189  ;2639   lc_rc[0e]  ; ...
U 1189,  0000,003C,0180,0A78,0010,118A  ;2640  alu_r[0f],clk.ubcc ; Check for no response
U 118A,  0001,013C,0180,0880,0082,108A  ;2641  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 108A,  0000,003C,0180,0800,0000,118B  ;2642 =0 j/check.adpt.code ; Check for a memory
U 108B,  0000,003C,0180,0800,0000,11AA  ;2643  j/find.mem.lp1  ; Try next tr
            ;2644 CHECK.ADPT.CODE:
```

```
U 118B, 0000,003C,1D80,0800,1404,718C  ;2645  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 118C, 0000,163C,0180,0800,0000,1008  ;2646  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,718D  ;2647 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,718E  ;2648  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11AA  ;2649  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11AA  ;2650  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7193  ;2651  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11AA  ;2652  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1197  ;2653  rc[0e]_ic,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1197  ;2654  rc[0e]_ic,j/found.ms780e ; MS780-E
           ;2655   ;
           ;2656   ; Found an MS780-A or -C. Set the starting address
           ;2657   ; to maximum.
           ;2658   ;
U 118D, 0818,0038,5D80,0800,0000,118F  ;2659 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 118E, 0818,0038,0580,0800,0000,118F  ;2660 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
           ;2661 MS780.COMMON:
U 118F, 0819,0030,7180,0800,0000,1190  ;2662  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1190, 0000,003C,01F8,0803,0080,D191  ;2663  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1191, 0D00,003C,0180,0800,0000,1192  ;2664  d_dal.sc  ; Put data in bits<29:14>
           ;2665  cache.p_d[long], ; Set the starting address to maximum
U 1192, 0000,003C,0180,A800,0000,11AA  ;2666       j/find.mem.lp1 ; and continue TR scan
           ;2667   ;
           ;2668   ; Found an MA780. Set the starting address to maximum
           ;2669   ;
           ;2670 MA780.MAX:
U 1193, 0818,0038,39F8,0803,0000,1194  ;2671  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1194, 0D00,003C,0180,0803,0000,1195  ;2672  d_dal.sc,va_va+4 ; Put in proper position
U 1195, 0000,003C,0180,0803,0000,1196  ;2673  va_va+4   ; Point to Port Invalidate Ctrl Register
           ;2674  cache.p_d[long], ; Set starting address to maximum
U 1196, 0000,003C,0180,A800,0000,11AA  ;2675       j/find.mem.lp1
           ;2676 ;
           ;2677 ; Found an MS780E
           ;2678 ;
           ;2679 FOUND.MS780E:
U 1197, 0000,003C,F9F0,2C00,0000,1198  ;2680  q_id[3e]  ; Set up to see how many MS780-E's
           ;2681  alu_q.xor.k[.3], ; Are there Maximum units?
U 1198, 0019,2020,0D80,0800,0010,1199  ;2682   clk.ubcc ; Check condition codes
U 1199, 0000,013C,0180,0800,0000,108C  ;2683  z?   ; Are we at maximum units for system
U 108C, 0000,003C,0180,0800,0000,119A  ;2684 =0 J/ms780e.tst  ; No go find how many units ther are
U 108D, 0B18,0038,C180,0800,0000,108E  ;2685  d_k[.ffff]  ; Yes set address to maximum
U 108E, 0000,003D,0180,0800,0000,11B2  ;2686 =0 call[set.start.addr] ; .....
U 108F, 0000,003C,0180,0800,0000,11AA  ;2687  j/find.mem.lp1  ; Go check to see if we are done
           ;2688 ;
           ;2689 MS780E.TST:
           ;2690  alu_rc[01],  ; Check "Found MS780E Flag"
U 119A, 0010,0038,0180,0908,0010,119B  ;2691   clk.ubcc ; Check condition codes
U 119B, 0810,0138,0180,0970,0000,1090  ;2692  z?,d_rc[0e]  ; Is this unit the first MS780E ?
           ;2693 =0 j/not.first.ms780e, ; No
U 1090, 0818,0038,C180,0800,0000,1094  ;2694   d_k[.ffff] ; Set starting address
U 1091, 0001,003C,0180,0988,0000,119C  ;2695  rc[01]_d  ; Yes put base address in rc[01]
U 119C, 0818,0038,7980,0800,0000,119D  ;2696  d_k[.30]  ; Set up SC to point to first unit
U 119D, 0019,0014,F580,0800,0082,119E  ;2697  alu_d+k[.a],sc_alu ; ...
U 119E, 0810,0038,0180,0908,0000,119F  ;2698  d_rc[01]  ; Put base address of unit in D
U 119F, 0000,003C,0180,3400,0000,11A0  ;2699  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11A0. 0F00.003C.0180.0800.0000.1092  ;2700  d_0   ; Prepare to set starting address to Zero
U 1092. 0000.003D.0180.0800.0000.11B2  ;2701 =0 call[set.start.addr] ; Go do it
        ;2702  j/find.mem.lp1,  ; Check to see if we are done
U 1093. 0003.003C.0180.09E8.0000.11AA  ;2703   rc[0d]_0 ; ....
        ;2704 =0
        ;2705 NOT.FIRST.MS780E:
U 1094. 0000.003D.0180.0800.0000.11B2  ;2706  call[set.start.addr] ; Go set starting address to maximum
U 1095. 0000.003C.F9F0.2C00.0000.11A1  ;2707  q_id[3e]  ; Get the number of MS780-Es found
U 11A1. 0819.2014.0580.0800.0000.11A2  ;2708  d_q+k[.1]  ; Increment this counter
U 11A2. 0000.003C.F980.3C00.0000.11A3  ;2709  id[3e]_d  ; Save the counter
U 11A3. 0018.0038.11C0.0800.0000.11A4  ;2710  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2711     ; ...were the I/O base addresses will be stored
U 11A4. 081D.2000.7980.0800.0000.11A5  ;2712  d_q-d,k[.30]  ; ... adjust pointer
U 11A5. 0819.0014.7980.0800.0000.11A6  ;2713  d_d+k[.30]  ; Point the SC to the ID register
U 11A6. 0000.003C.F580.0800.0000.11A7  ;2714  k[.a]   ; ...to receive I/O base address
U 11A7. 0019.0014.F580.0800.0082.11A8  ;2715  alu_d+k[.a],sc_alu ; ...
U 11A8. 0810.0038.0180.0970.0000.11A9  ;2716  d_rc[0e]  ; Get base address to d
U 11A9. 0000.003C.0180.3400.0000.11AA  ;2717  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2718   ;
        ;2719   ; Try next tr
        ;2720   ;
        ;2721 FIND.MEM.LP1:
U 11AA. 0818.0014.0580.0A80.0000.11AB  ;2722  r[0]_ia+k[.1],d_alu ; Increment TR level
        ;2723  alu_d.and.k[.f],
U 11AB. 0019.0034.6180.0800.0010.11AC  ;2724      clk.ubcc  ; Check if all done
U 11AC. 0000.013C.0180.0800.0000.1096  ;2725  z?   ; Branch if yes
U 1096. 0000.003C.0180.0800.0000.1086  ;2726 =0 j/find.mem.loop  ; Try next TR
U 1097. 0810.0038.0180.0908.0010.11AD  ;2727  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11AD. 0000.013C.0180.0800.0000.1098  ;2728  z?   ; Check condition codes
U 1098. 0001.003E.0180.09F0.0000.0002  ;2729 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1099. 0000.003E.0180.0800.0000.0001  ;2730  return1   ; No MS780E found
            ;2731
```

```
            ;2732  .PAGE
            ;2733  .TOC "   Generate IO Address"
            ;2734 ;++
            ;2735 ; FUNCTIONAL DESCRIPTION:
            ;2736 ;
            ;2737 ; This routine is used to generate an SBI Configuration Register
            ;2738 ; address from a TR level.
            ;2739 ;
            ;2740 ; CALLING CONSTRAINT:
            ;2741 ;
            ;2742 ; =0
            ;2743 ;
            ;2744 ; INPUTS:
            ;2745 ;
            ;2746 ; D - Contains TR level
            ;2747 ;
            ;2748 ; OUTPUTS:
            ;2749 ;
            ;2750 ; D - Contains SBI IO address
            ;2751 ;--
            ;2752 GENERATE.IO:
U 11AE, 0000,003C,89F8,0800,0084,71AF  ;2753  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11AF, 0D00,003C,8D80,0800,0084,71B0  ;2754  d_del.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11B0, 0000,003C,0980,0800,0084,B1B1  ;2755  sc_sc-k[.2]  ; insert bit 29
U 11B1, 0801,001E,0180,0800,0000,0001  ;2756  d_d.ornot.mask,return1 ; and return
            ;2757
```

```
            ;2758  .PAGE
            ;2759  .TOC "   Set Starting Address"
            ;2760 ;**
            ;2761 ; FUNCTIONAL DESCRIPTION:
            ;2762 ;
            ;2763 ; This routine is used to set the starting address of the
            ;2764 ; memory to the specified value.
            ;2765 ;
            ;2766 ; CALLING CONSTRAINT:
            ;2767 ;
            ;2768 ; =0
            ;2769 ;
            ;2770 ; INPUTS:
            ;2771 ;
            ;2772 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2773 ;     (B Register Image divided by 2**16)
            ;2774 ; rc[0] - Contains Address of Register A
            ;2775 ;
            ;2776 ; OUTPUTS:
            ;2777 ;
            ;2778 ; d - Contains aligned starting address (B Register Image)
            ;2779 ;
            ;2780 ; SIDE EFFECTS:
            ;2781 ;
            ;2782 ; d,q,va, and sc are destroyed
            ;2783 ;--
            ;2784 SET.START.ADDR:
U 11B2, 0010,0038,6580,0970,0284,71B3  ;2785  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11B3, 0000,003C,01F8,0803,0000,11B4  ;2786  va_va+4,q_0  ; Set address to Register B
            ;2787  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11B4, 0D00,003C,0980,0800,0084,B1B5  ;2788       sc_sc-k[.2]  ; Setup to insert bit 14
U 11B5, 0801,001C,0180,0800,0000,11B6  ;2789  d_d.ornot.mask  ; Insert Write Enable bit
U 11B6, 0000,003E,0180,A800,0000,0001  ;2790  cache.p_d[long],return1 ; Set the starting address and return
            ;2791
            ;2792
```

```
                ;2793  .PAGE
                ;2794  .TOC "   CONFIGURE MS780-E/H"
                ;2795 ;*****************************************************************
                ;2796 ;++
                ;2797 ;
                ;2798 ; Functional Description:
                ;2799 ; ----------------------
                ;2800 ;
                ;2801 ;      This  subroutine  will  be  used  by  tests  that   do   not
                ;2802 ; specifically  check  the  memory, but  make use of it to execute.
                ;2803 ; Its purpose is to find a MS780-E and  configure  it  properly  so
                ;2804 ; that  if  a  Read/Write operation will take place no false errors
                ;2805 ; will be logged due to misconfigured memory.
                ;2806 ;
                ;2807 ; Calling Constraint: =00
                ;2808 ; ------------------
                ;2809 ;
                ;2810 ;
                ;2811 ; Inputs:
                ;2812 ; ------
                ;2813 ;
                ;2814 ; RC[0E] - I/O BASE OF MS780-E/H
                ;2815 ;
                ;2816 ;
                ;2817 ; Outputs:
                ;2818 ; -------
                ;2819 ;
                ;2820 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
                ;2821 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
                ;2822 ;      OR IF NO MS780-E/Hs WERE FOUND
                ;2823 ;      ON THE SBI
                ;2824 ;
                ;2825 ; Side Effects:
                ;2826 ; ------------
                ;2827 ;
                ;2828 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
                ;2829 ;
                ;2830 ;
                ;2831 ;
                ;2832 ;--
                ;2833 ;*****************************************************************
                ;2834 CONFIG.MS780E:
U 11B7, 0818,0038,0D80,0800,0000,11B8  ;2835  D_K[.3]    ; Set Flag for the Fail Chain
U 11B8, 0001,003C,0180,09E8,0000,1018  ;2836  RC[0D]_D    ; ...
U 1018, 0000,003D,0180,0800,0000,1084  ;2837 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1019, 0000,003D,0980,0800,0084,7058  ;2838  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 101A, 0000,003C,0180,0800,0000,1020  ;2839  NOP     ; OK. Found at least one MS780-E/H
                ;2840 =
                ;2841 CONFIG.RETRY:    ; Entry point for the case in which the
                ;2842       ; ...first MS780-E/H could not be
                ;2843       ; ...properly configured
U 1020, 0000,003D,0180,0800,0000,1004  ;2844 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1021, 0000,003C,0180,0800,0000,1024  ;2845  J/CNFG.LMIS   ; Lower controller misconfigured
U 1022, 0000,003E,0180,0800,0000,0001  ;2846  RETURN1    ; OK. Lower Controller is configured
                ;2847 =
```

```
            ;2848 =00
            ;2849 CNFG.LMIS:
U 1024, 0000,003D,0180,0800,0000,1010  ;2850  CALL[SELECT.UPPER]  ; Select the upper controller
U 1025, 0000,003C,0180,0800,0000,1028  ;2851  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 1026, 0000,003E,0180,0800,0000,0001  ;2852  RETURN1    ; OK. Upper Controller is configured
            ;2853 =
            ;2854 =00
            ;2855 CNFG.UMIS:
U 1028, 0000,003D,0180,0800,0000,11BA  ;2856  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2857       ; ...MS780-E/H found so go and see
            ;2858       ; ...if there are any more
U 1029, 0000,003C,0180,0800,0000,11B9  ;2859  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2860       ; ...try to configure it
U 102A, 0818,0038,0D80,0800,0000,102B  ;2861  D_K[.3]    ; Set Flag for the Fail Chain
U 102B, 0001,003C,0180,09E8,0000,11B9  ;2862  RC[0D]_D    ; ...
U 11B9, 0000,003D,0980,0800,0084,7058  ;2863  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2864       ; ...abort the test
            ;2865 =
            ;2866
```

```
            ;2867  .PAGE
            ;2868  .TOC "    ENABLE NEXT MS780-E"
            ;2869 ;**
            ;2870 ; FUNCTIONAL DESCRIPTION:
            ;2871 ;
            ;2872 ;           This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2873 ;     They will be tested  separately, according to the TR level at which
            ;2874 ;     they are located,  starting with the one  at the lowest level first.
            ;2875 ;           When  a test has finished  with the first unit, it will have to call
            ;2876 ;     this specific  routine  to see  if any other MS780-E unit is present
            ;2877 ;     on the SBI. If more units are  present, the first one  will be  dis-
            ;2878 ;     abled  by  setting  its  starting  address to maximum, the next unit
            ;2879 ;     will be  enabled  by  setting its starting address to 0 and the test
            ;2880 ;     will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2881 ;     for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2882 ;     ID registers 3A through 3D and  a count of the Total number of units
            ;2883 ;     found - 1 in register 3E. In this subroutine the SC register is used
            ;2884 ;     as  a  pointer  to  ID  registers 3A through 3D.
            ;2885 ;
            ;2886 ; CALLING CONSTRAINT:
            ;2887 ;
            ;2888 ;  =00
            ;2889 ;
            ;2890 ;--
            ;2891 ENABLE.NEXT.MS780E:
U 11BA, 0000,003C,F9F0,2C00,0000,11BB  ;2892  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2893  ALU_Q,    ; Check to see if it is zero
U 11BB, 0001,203C,0180,0800,0010,11BC  ;2894     CLK.UBCC  ; Check Condition Codes
U 11BC, 0000,013C,0180,0800,0000,109A  ;2895  Z?    ; ...for zero
U 109A, 0000,003C,0180,0800,0000,11BD  ;2896 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 109B, 0000,003E,0180,0800,0000,0002  ;2897  RETURN2    ; Zero No more units to test
            ;2898 ENABLE.NEXT:
U 11BD, 0818,0038,C180,0800,0000,109C  ;2899  D_K[.FFFF]  ; Load D with Maximum Starting address
U 109C, 0000,003D,0180,0800,0000,11B2  ;2900 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 109D, 0818,0038,7980,0800,0000,11BE  ;2901  D_K[.30]  ; Prepare to point to the ID register holding
            ;2902      ; ...the I/O Base address of the next MS780-E
U 11BE, 0819,0014,1180,0800,0000,11BF  ;2903  D_D+K[.4]  ; ...
U 11BF, 0000,003C,F580,0800,0000,11C0  ;2904  K[.A]    ; ...
U 11C0, 0819,0014,F580,0800,0000,11C1  ;2905  D_D+K[.A]  ; ...
U 11C1, 0000,003C,F9F0,2C00,0000,11C2  ;2906  Q_ID[3E]  ; Get MS780-E Counter
            ;2907  D_D-Q,    ; D now holds the pointer to the ID register
U 11C2, 081D,0000,0180,0800,0082,11C3  ;2908    SC_ALU  ; ...
U 11C3, 0000,003C,01F0,2400,0000,11C4  ;2909  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11C4, 0001,203C,0180,09F0,0000,11C5  ;2910  RC[0E]_Q  ; Save it also in RC[0E]
U 11C5, 0F03,003C,0180,09E8,0000,109E  ;2911  RC[0D]_0,D_0  ; Set starting address to zero
U 109E, 0000,003D,0180,0800,0000,11B2  ;2912 =0 CALL[SET.START.ADDR] ; ....
U 109F, 0F00,003C,F9F0,2C00,0000,11C6  ;2913  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11C6, 081D,2008,0180,0800,0000,11C7  ;2914  D_Q-D-1    ; Subtract 1 from total
U 11C7, 0000,003C,F980,3C00,0000,10A0  ;2915  ID[3E]_D  ; Adjust the pointer
U 10A0, 0000,003D,0180,0800,0000,1152  ;2916 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10A1, 0000,003E,0180,0800,0000,0001  ;2917  RETURN1    ; Tell caller another unit was found
            ;2918
```

```
;2919  .PAGE "TEST 1D5 CACHE/SBI INVALIDATE TEST "
;2920 ;=============================================================================
;2921 ;**
;2922 ;
;2923 ; SBI001.MIC
;2924 ;
;2925 ; CACHE/SBI INVALIDATE TEST
;2926 ;
;2927 ; TEST DESCRIPTION
;2928 ;
;2929 ; CHECK TO SEE THE ENABLE SBI INVALIDATE
;2930 ; AND FORCE SBI INVALIDATE WORK. WILL CHECK THE
;2931 ; HIT/MISS CACHE BIT IN MAINTAINANCE REGISTER TO
;2932 ; SEE THAT THE CACHE WAS INVALIDATED. THE ADDRESS WILL
;2933 ; REMAIN CONSTANT DURING THE TEST. BOTH GROUPS
;2934 ; WILL BE FORCED AND CHECKED.
;2935 ;
;2936 ;
;2937 ;
;2938 ; SUBTEST 01 ==================================================
;2939 ;
;2940 ;  MAINT REG: BIT 21=0 ENABLE SBI INVALIDATE
;2941 ;       BIT 22=1 FORCE SBI INVALIDATE
;2942 ;
;2943 ;   SEE THAT THE CACHE IS NOT INVALIDATE
;2944 ;   ON CPU WRITE HITS
;2945 ;
;2946 ; SUBTEST 02 ==================================================
;2947 ;
;2948 ;  MAINT REG BIT 21=1 ENABLE SBI INVALIDATE
;2949 ;     BIT 22=0 FORCE SBI INVALIDATE
;2950 ;
;2951 ;   SEE THAT THE CACHE IS NOT INVALIDATE
;2952 ;   ON CPU WRITE HITS
;2953 ;
;2954 ;
;2955 ; SUBTEST 03 ==================================================
;2956 ;
;2957 ;  MAINT REG BIT 21=1 ENABLE SBI INVALIDATE
;2958 ;     BIT 22=1 FORCE SBI INVALIDATE
;2959 ;     SEE THAT THE CACHE IS INVALIDATED
;2960 ;     ON CPU WRITE HITS
;2961 ;
;2962 ;
;2963 ;
;2964 ;  NOTE:
;2965 ;  -----
;2966 ; RUN THE CACHE TESTS PREVIOUS TO THIS
;2967 ; TEST TO ASSURE THAT THE CACHE HIT/MISS
;2968 ; BITS WORK PROPERLY, I.E. CACHE DOES NOT RETURN
;2969 ; DATA ON MISS.
;2970 ;
;2971 ;
;2972 ; *** IMPORTANT ***
;2973 ;
```

```
;2974 ; THIS TEST WILL ONLY OPERATE ON A MS780-E/H.
;2975 ; IT WILL NOT TEST ALL MS780-E/Hs ON THE SBI BUT
;2976 ; ONLY THE ONE AT TR LEVEL 1 (OR THE FIRST MS780-E/H
;2977 ; THAT CAN BE CONFIGURED).
;2978 ;
;2979 ; LOGIC DESCRIPTION
;2980 ;
;2981 ; SBL(M8218) AND SBH(M8219)
;2982 ;  MAINT REGISTER BITS 22,21(SBH)
;2983 ;  WRITE INVALIDATE LOGIC FOR CACHE (SBH)
;2984 ;  FORCE CACHE GROUP REPLACEMENT(MAINT REG,BITS 13,14) BITS
;2985 ; CACHE ADDRESS MATRIX FOR HIT DETECTION
;2986 ;
;2987 ; ERROR LOGOUT
;2988 ;
;2989 ; DATA: CACHE GROUP CURRENTLY BEING FORCED
;2990 ;       =0 GROUP 0
;2991 ;       =1 GROUP 1
;2992 ;  TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
;2993 ;  UNDEFINED
;2994 ;  UNDEFINED
;2995 ;  UNDEFINED
;2996 ;  UNDEFINED
;2997 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;2998 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;2999 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3000 ;  INTERRUPT FLAG:
;3001 ;    0=>NO INTERRUPT OCCURRED
;3002 ;    1=>AN INTERRUPT OCCURRED
;3003 ;
;3004 ; SYNC POINT DESCRIPTION
;3005 ;
;3006 ;--
;3007 ;*******************************************************************************
;3008 ;
;3009 ;  SCRATCH PAD
;3010 ; RA
;3011 ; 0 CACHE HIT CONSTANT ^X600
;3012 ; 1 GROUP FLAG =0 GROUP 0
;3013 ;     =1 GROUP 1
;3014 ; F U-TRAP FLAG
;3015 ;
;3016 ; RC
;3017 ; 0 CONFIG REG A ADDRESS ^X20002000
;3018 ; 1 CONFIG REG B ADDRESS ^X20002004
;3019 ; 2 CONFIG REG C ADDRESS ^X20002008
;3020 ; F #ARG TO CONSOLE  ^X1
;3021 ;
;3022 ;===============================================================================
;3023 ;
;3024 ;
;3025 =0
U 10A2, 0000,003D,0180,0800,0000,104F  ;3026 S01000: NEWTST
U 10A3, 0018,0038,F580,09F8,0000,10A4  ;3027  RC[0F]_K[.A]
U 10A4, 0000,003D,0180,0800,0000,11B7  ;3028 =0 CALL[CONFIG.MS780E] ; Try to configure the MS780-E memory
```

```
U 10A5, 0000.003C.0180.0800.0000.11C8  ;3029  NOP
U 11C8, 0818.0038.D5F8.0800.0000.11C9  ;3030  D_K[.6],Q_0  ;R 0
        ;3031
U 11C9, 0000.003C.0180.0800.0084.71CA  ;3032  SC_K[.8]
U 11CA, 0D00.003C.0180.0800.0000.11CB  ;3033  D_DAL.SC
U 11CB, 0001.003C.0180.0A80.0000.10A6  ;3034  R[0]_D
        ;3035 ;
        ;3036 ;================================================================================
        ;3037 ;*
        ;3038 ;SUBTEST 01
        ;3039 ;
        ;3040 ; TURN THE CACHE ON, VALIDATE, BOTH UPPER AND
        ;3041 ; LOWER,FORCE MAINT BIT 21,22 TO 0,1 RESPECTIVELY
        ;3042 ; AND DO A WRITE OPERATION.  READ AND
        ;3043 ; SEE A HIT, IE. CACHE WAS NOT INVALIDATE
        ;3044 ; BY THE WRITE OPERATION WITH FORCE INVALIDATE
        ;3045 ; SET (BIT 23) AND NOT ENABLE SBI INVALIDATE (BIT 22)
        ;3046 ;-
        ;3047 ;================================================================================
        ;3048 ;
U 10A6, 0000.003D.0180.0800.0000.1134  ;3049 =0 SUBTEST
U 10A7, 0003.003C.0180.0A88.0000.10A8  ;3050  R[1]_0     ;SET GROUP FLAG
U 10A8, 0000.003D.0180.0800.0000.112F  ;3051 =0 ERLOOP
U 10A9, 0000.003C.0180.0800.0000.10AA  ;3052 S01010: NOP
U 10AA, 0000.003D.0180.0800.0000.1159  ;3053 =0 CALL[INIT]
U 10AB, 0018.0038.1980.0800.0200.11CC  ;3054  VA_K[ZERO]    ;LOAD ADDRESS
U 11CC, 0818.0038.4580.0800.0000.11CD  ;3055  D_K[.8000]    ;SET FORCE REPLACE CONSTANT
U 11CD, 060C.003C.0180.0A08.0010.11CE  ;3056  ALU_R[1],CLK.UBCC,D_D.RIGHT ;TEST FLAG, SHIFT DATA
U 11CE, 0003.013C.0180.09F0.0000.10AC  ;3057  Z?,RC[0E]_0    ;SET CONSOLE FLAG
U 10AC, 0618.0038.0580.09F0.0000.10AD  ;3058 =0 D_D.RIGHT,RC[0E]_K[.1]  ;SET FOR GROUP 1
U 10AD, 0000.003C.7580.3C00.0000.11CF  ;3059  ID[MAINT]_D    ;WRITE THE MAINTENANCE REGISTER
U 11CF, 0000.003C.0180.9800.0000.11D0  ;3060  MCT/VALIDATE    ;VALIDATE CACHE
U 11D0, 0000.003C.0180.0803.0000.11D1  ;3061  VA_VA+4
U 11D1, 0000.003C.0180.9800.0000.11D2  ;3062  MCT/VALIDATE
U 11D2, 0018.0038.1980.0800.0200.11D3  ;3063  VA_K[ZERO]    ;RESET ADDRESS
U 11D3, 0818.0038.11F8.0800.0000.11D4  ;3064  D_K[.4],Q_0    ;^X400000 TO MAINT REG
U 11D4, 0000.003C.2180.0800.0084.71D5  ;3065  SC_K[.14]    ;I.E. FORCE INVALIDATE
U 11D5, 0D00.003C.0180.0800.0000.11D6  ;3066  D_DAL.SC    ;NO INVALIDATE ENABLE
U 11D6, 0000.003C.7580.3C00.0000.11D7  ;3067  ID[MAINT]_D
U 11D7, 0000.003C.0180.A800.0000.10AE  ;3068  MCT/WRITE.P    ;DO THE WRITE OPERATION
        ;3069 =0 D_K[.6],
U 10AE, 0818.0039.D580.0800.0000.1157  ;3070   CALL[WAIT.LOOP]
U 10AF, 0000.003C.0180.0800.0000.102C  ;3071  NOP
U 102C, 0000.003D.0180.0800.0000.1060  ;3072 =00 CALL[NOINTR]
U 102D, 0000.003D.0180.0800.0000.1130  ;3073  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 102E, 0000.003C.0180.0800.0000.11D8  ;3074  NOP
U 11D8, 0000.003C.0180.C800.0000.1030  ;3075 = MCT/READ.P
U 1030, 0000.003D.0180.0800.0000.1060  ;3076 =00 CALL[NOINTR]
U 1031, 0000.003D.0180.0800.0000.1130  ;3077  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1032, 0000.003C.75F0.2C00.0000.11D9  ;3078  Q_ID[MAINT]    ;SEE A HIT IN CACHE
U 11D9, 000D.2034.0180.0A00.0010.11DA  ;3079 = ALU_Q[AND]R[0],CLK.UBCC
U 11DA, 0000.013C.0180.0800.0000.10B0  ;3080  Z?
U 10B0, 0000.003C.0180.0800.0000.11DB  ;3081 =0 J/S01090    ;TEST PASSED, CACHE HIT
U 10B1, 0000.003D.0180.0800.0000.1130  ;3082  ERROR1     ;TEST FAILED,CACHE MISS
U 11DB, 0000.003C.0180.0A08.0010.11DC  ;3083 S01090: ALU_R[1],CLK.UBCC  ;CHECK GROUP FLAG
```

```
U 11DC, 0000.013C.0180.0800.0000.10B2  ;3084  Z?
U 10B2, 0000.003C.0180.0800.0000.11DD  ;3085 =0 J/S01100   ;IT'S SET END TEST
U 10B3, 0018.0038.0580.0A88.0000.10A9  ;3086  R[1]_K[.1],J/S01010  ;SET IT AND REPEAT
          ;3087 ;
          ;3088 ;=================================================================================
          ;3089 ;*
          ;3090 ;SUBTEST 02
          ;3091 ;
          ;3092 ; SEE THAT THE CACHE IS NOT INVALIDATED
          ;3093 ; WHEN INVALIDATE IS ENABLED (MAINT REG BIT 21=1) AND THE
          ;3094 ; CPU DOES A WRITE, FORCE INVALIDATE (BIT 22=0)
          ;3095 ; IS NOT ACTIVE
          ;3096 ;-
          ;3097 ;=================================================================================
          ;3098 ;
U 11DD, 0000.003C.0180.0800.0000.10B4  ;3099 S01100: NOP
U 10B4, 0000.003D.0180.0800.0000.1134  ;3100 =0 SUBTEST
U 10B5, 0003.003C.0180.0A88.0000.10B6  ;3101  R[1]_0    ;SET GROUP FLAG
U 10B6, 0000.003D.0180.0800.0000.112F  ;3102 =0 ERLOOP
U 10B7, 0000.003C.0180.0800.0000.10B8  ;3103 S01110: NOP
U 10B8, 0000.003D.0180.0800.0000.1159  ;3104 =0 CALL[INIT]
U 10B9, 0018.0038.1980.0800.0200.11DE  ;3105  VA_K[ZERO]   ;SET ADDRESS
U 11DE, 0818.0038.4580.0800.0000.11DF  ;3106  D_K[.8000]   ;SET FORCE REPLACE CONSTANT
U 11DF, 0600.003C.0180.0A08.0010.11E0  ;3107  ALU_R[1],CLK.UBCC,D_D.RIGHT ;TEST FLAG, SHIFT DATA
U 11E0, 0003.013C.0180.09F0.0000.10BA  ;3108  Z?,RC[0E]_0   ;SET CONSOLE FLAG
U 10BA, 0618.0038.0580.09F0.0000.10BB  ;3109 =0 D_D.RIGHT,RC[0E]_K[.1]  ;SET FOR GROUP 1
U 10BB, 0000.003C.7580.3C00.0000.11E1  ;3110  ID[MAINT]_D   ;WRITE THE MAINTENANCE REGISTER
U 11E1, 0000.003C.0180.9800.0000.11E2  ;3111  MCT/VALIDATE   ;VALIDATE CACHE ENTRIES
U 11E2, 0000.003C.0180.0803.0000.11E3  ;3112  VA_VA+4
U 11E3, 0000.003C.0180.9800.0000.11E4  ;3113  MCT/VALIDATE
U 11E4, 0018.0038.1980.0800.0200.11E5  ;3114  VA_K[ZERO]   ;RESET ADDRESS
U 11E5, 0818.0038.09F8.0800.0000.11E6  ;3115  D_K[.2],Q_0   ;^X200000 TO MAINT KEY
U 11E6, 0000.003C.2180.0800.0084.71E7  ;3116  SC_K[.14]   ;I.E ENABLE INVALIDATE
U 11E7, 0D00.003C.0180.0800.0000.11E8  ;3117  D_DAL.SC   ;NO FORCE INVALIDATE
U 11E8, 0000.003C.7580.3C00.0000.11E9  ;3118  ID[MAINT]_D
U 11E9, 0000.003C.0180.A800.0000.10BC  ;3119  MCT/WRITE.P   ;DO THE WRITE OPERATION
          ;3120 =0 D_K[.6],
U 10BC, 0818.0039.D580.0800.0000.1157  ;3121   CALL[WAIT.LOOP]
U 10BD, 0000.003C.0180.0800.0000.1034  ;3122  NOP
U 1034, 0000.003D.0180.0800.0000.1060  ;3123 =00 CALL[NOINTR]
U 1035, 0000.003D.0180.0800.0000.1130  ;3124  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1036, 0000.003C.0180.0800.0000.11EA  ;3125  NOP
U 11EA, 0000.003C.0180.C800.0000.1038  ;3126 = MCT/READ.P   ;READ FOR CACHE HIT
U 1038, 0000.003D.0180.0800.0000.1060  ;3127 =00 CALL[NOINTR]
U 1039, 0000.003D.0180.0800.0000.1130  ;3128  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 103A, 0000.003C.75F0.2C00.0000.11EB  ;3129  Q_ID[MAINT]   ;CHECK FOR CACHE HIT
U 11EB, 000D.2034.0180.0A00.0010.11EC  ;3130 = ALU_Q[AND]R[0],CLK.UBCC
U 11EC, 0000.013C.0180.0800.0000.10BE  ;3131  Z?
U 10BE, 0000.003C.0180.0800.0000.11ED  ;3132 =0 J/S01190   ;CACHE HIT,CONTINUE
U 10BF, 0000.003D.0180.0800.0000.1130  ;3133  ERROR1    ;MISS, ENTRY WAS INVALIDATED
U 11ED, 0000.003C.0180.0A08.0010.11EE  ;3134 S01190: ALU_R[1],CLK.UBCC  ;CHECK GROUP FLAG
U 11EE, 0000.013C.0180.0800.0000.10C0  ;3135  Z?
U 10C0, 0000.003C.0180.0800.0000.11EF  ;3136 =0 J/S01200   ;IT'S SET END TEST
U 10C1, 0018.0038.0580.0A88.0000.10B7  ;3137  R[1]_K[.1],J/S01110  ;SET IT AND REPEAT
          ;3138 ;
```

```
          ;3139 ;=================================================================================
          ;3140 ;*
          ;3141 ;SUBTEST 03
          ;3142 ;
          ;3143 ; SEE THAT THE CACHE IS INVALIDATED
          ;3144 ; WHEN BIT 21,22 ARE 1,1 RESPECTIVELY. THIS ENABLE
          ;3145 ; INVALIDATE, AND FORCES INVALIDATE ALSO
          ;3146 ; I.E. AFTER THE WRITE READ AND SEE MISS
          ;3147 ;-
          ;3148 ;=================================================================================
          ;3149 ;
U 11EF, 0000.003C.0180.0800.0000.10C2  ;3150 S01200: NOP
U 10C2, 0000.003D.0180.0800.0000.1134  ;3151 =0 SUBTEST
U 10C3, 0003.003C.0180.0A88.0000.10C4  ;3152  R[1]_0     ;SET GROUP FLAG
U 10C4, 0000.003D.0180.0800.0000.112F  ;3153 =0 ERLOOP
U 10C5, 0000.003C.0180.0800.0000.10C6  ;3154 S01210: NOP
U 10C6, 0000.003D.0180.0800.0000.1159  ;3155 =0 CALL[INIT]
U 10C7, 0018.0038.1980.0800.0200.11F0  ;3156  VA_K[ZERO]    ;SET ADDRESS
U 11F0, 0818.0038.4580.0800.0000.11F1  ;3157  D_K[.8000]    ;SET FORCE REPLACE CONSTANT
U 11F1, 0600.003C.0180.0A08.0010.11F2  ;3158  ALU_R[1],CLK.UBCC,D_D.RIGHT ;TEST FLAG, SHIFT DATA
U 11F2, 0003.013C.0180.09F0.0000.10C8  ;3159  Z?,RC[0E]_0    ;SET CONSOLE FLAG
U 10C8, 0618.0038.0580.09F0.0000.10C9  ;3160 =0 D_D.RIGHT,RC[0E]_K[.1]  ;SET FOR GROUP 1
U 10C9, 0000.003C.7580.3C00.0000.11F3  ;3161  ID[MAINT]_D    ;WRITE THE MAINTENANCE REGISTER
U 11F3, 0000.003C.0180.9800.0000.11F4  ;3162  MCT/VALIDATE     ;VALIDATE THE CACHE
U 11F4, 0000.003C.0180.0803.0000.11F5  ;3163  VA_VA+4
U 11F5, 0000.003C.0180.9800.0000.11F6  ;3164  MCT/VALIDATE
U 11F6, 0018.0038.1980.0800.0200.11F7  ;3165  VA_K[ZERO]    ;RESET TIMER
U 11F7, 0818.0038.D5F8.0800.0000.11F8  ;3166  D_K[.6],Q_0    ;^X600000 TO MAINT REG
U 11F8, 0000.003C.2180.0800.0084.71F9  ;3167  SC_K[.14]    ;I.E. ENABLE INVALIDATE
U 11F9, 0D00.003C.0180.0800.0000.11FA  ;3168  D_DAL.SC    ;FORCE INVALIDATE
U 11FA, 0000.003C.7580.3C00.0000.11FB  ;3169  ID[MAINT]_D
U 11FB, 0000.003C.0180.A800.0000.10CA  ;3170  MCT/WRITE.P
          ;3171 =0 D_K[.6],
U 10CA, 0818.0039.D580.0800.0000.1157  ;3172   CALL[WAIT.LOOP]
U 10CB, 0000.003C.0180.0800.0000.103C  ;3173  NOP
U 103C, 0000.003D.0180.0800.0000.1060  ;3174 =00 CALL[NOINTR]
U 103D, 0000.003D.0180.0800.0000.1130  ;3175  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 103E, 0000.003C.0180.0800.0000.11FC  ;3176  NOP
U 11FC, 0000.003C.0180.C800.0000.1040  ;3177 = MCT/READ.P    ;READ TO CHECK FOR MISS
U 1040, 0000.003D.0180.0800.0000.1060  ;3178 =00 CALL[NOINTR]
U 1041, 0000.003D.0180.0800.0000.1130  ;3179  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1042, 0000.003C.75F0.2C00.0000.11FD  ;3180  Q_ID[MAINT]
U 11FD, 000D.2034.0180.0A00.0010.11FE  ;3181 = ALU_Q[AND]R[0],CLK.UBCC
U 11FE, 0000.013C.0180.0800.0000.10CC  ;3182  Z?
U 10CC, 0000.003D.0180.0800.0000.1130  ;3183 =0 ERROR1     ;HIT OCCURED, CACHE WAS
          ;3184       ;NOT INVALIDATED
U 10CD, 0000.003C.0180.0A08.0010.11FF  ;3185 S01290: ALU_R[1].CLK.UBCC  ;CHECK GROUP FLAG
U 11FF, 0000.013C.0180.0800.0000.10CE  ;3186  Z?
U 10CE, 0000.003C.0180.0800.0000.1200  ;3187 =0 J/S01END    ;IT'S SET END TEST
U 10CF, 0018.0038.0580.0A88.0000.10C5  ;3188  R[1]_K[.1],J/S01210  ;SET IT AND REPEAT
U 1200, 0000.003C.0180.0800.0000.10D0  ;3189 S01END: NOP     ; ALL DONE.
          ;3190
```

```
                ;3191  .PAGE "TEST 1D6 INTERRUPT DISABLE TESTS"
                ;3192 ;
                ;3193 ;*****************************************************************************
                ;3194 ;++
                ;3195 ;
                ;3196 ; INTERRUPT DISABLE TESTS
                ;3197 ;
                ;3198 ; This test will check the ability of disabling FAULT interrupts.
                ;3199 ; The following cases will be tested:
                ;3200 ;
                ;3201 ; FAULT INTERRUPT DISABLE
                ;3202 ;
                ;3203 ; SILO INTERRUPTS DISABLE
                ;3204 ;
                ;3205 ; CRD/RDS INTERRUPTS DISABLE
                ;3206 ;
                ;3207 ; The test is divided in subtests as follows:
                ;3208 ;
                ;3209 ; SUBTEST 1 - FAULT INTERRUPT DISABLE
                ;3210 ;
                ;3211 ; SUBTEST 2 - SILO INTERRUPT DISABLE
                ;3212 ;
                ;3213 ; SUBTEST 3 - CRD/RDS INTERRUPT DISABLE
                ;3214 ;
                ;3215 ;--
                ;3216 ;*****************************************************************************
                ;3217 ;
                ;3218 =0
U 10D0. 0000.003D.0180.0800.0000.104F  ;3219 INTDIS: NEWTST    ; Initialize subtest number
U 10D1. 001B.0010.F580.09F8.0000.10D2  ;3220  RC[0F]_ALU,ALU_0+K[.A]+1 ; Set arguments to console to x0B
U 10D2. 0000.003D.0180.0800.0000.11B7  ;3221 =0 CALL[CONFIG.MS780E]  ; Configure MS780-Es if any
U 10D3. 0000.003C.0180.0800.0000.10D4  ;3222  NOP
                ;3223
                ;3224 ;
                ;3225 ;=============================================================================
                ;3226 ;+
                ;3227 ;SUBTEST  1
                ;3228 ;
                ;3229 ;   CHECK FAULT INTERRUPT DISABLE
                ;3230 ;   ID=1B,"FAULT", BIT #18
                ;3231 ;
                ;3232 ; ERROR LOGOUT
                ;3233 ;
                ;3234 ; DATA: TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
                ;3235 ;  UNDEFINED
                ;3236 ;  UNDEFINED
                ;3237 ;  UNDEFINED
                ;3238 ;  UNDEFINED
                ;3239 ;  UNDEFINED
                ;3240 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
                ;3241 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
                ;3242 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;3243 ;  INTERRUPT FLAG:
                ;3244 ;   0=>NO INTERRUPT OCCURRED
                ;3245 ;   1=>AN INTERRUPT OCCURRED
```

```
            ;3246 ;
            ;3247 ;-
            ;3248 ;================================================================================
            ;3249 ;
U 10D4. 0000.003D.0180.0800.0000.1134  ;3250 =0 SUBTEST
U 10D5. 0000.003C.0180.0800.0000.10D6  ;3251  NOP
U 10D6. 0000.003D.0180.0800.0000.112F  ;3252 =0 ERLOOP
U 10D7. 0000.003C.0180.0800.0000.10D8  ;3253  NOP
U 10D8. 0000.003D.0180.0800.0000.1159  ;3254 =0 CALL[INIT]
U 10D9. 0F00.003C.0180.0800.0000.1201  ;3255  D_0     ;SEE FAULT BIT CLEAR
U 1201. 0000.003C.6D80.3C00.0000.10DA  ;3256  ID[FAULT]_D
U 10DA. 0000.003D.F580.0800.0084.7138  ;3257 =0 SETRCF[.A]
U 10DB. 0000.003C.6DF0.2C00.0000.1202  ;3258  Q_ID[FAULT]
U 1202. 0818.0038.4580.0800.0000.1203  ;3259  D_K[.8000]
U 1203. 0100.003C.0180.0800.0000.1204  ;3260  D_D.LEFT2
U 1204. 0500.003C.0180.0800.0000.1205  ;3261  D_D.LEFT
U 1205. 001D.0034.0180.0800.0010.1206  ;3262  ALU_D[AND]Q,CLK.UBCC  ;DID IT CLEAR?
U 1206. 0000.013C.0180.0800.0000.10DC  ;3263  Z?
U 10DC. 0000.003D.0180.0800.0000.1130  ;3264 =0 ERROR1    ;FAULT FAULT INTERRUPT ENABLE
            ;3265       ;   DIDN'T CLEAR
U 10DD. 0000.003C.6D80.3C00.0000.1207  ;3266  ID[FAULT]_D   ;SET BIT #18
U 1207. 0000.003C.6DF0.2C00.0000.1208  ;3267  Q_ID[FAULT]
U 1208. 001D.0034.0180.0800.0010.1209  ;3268  ALU_D[AND]Q,CLK.UBCC
U 1209. 0000.013C.0180.0800.0000.10DE  ;3269  Z?
U 10DE. 0000.003C.0180.0800.0000.120A  ;3270 =0 J/SA0010   ;YES IT'S SET
U 10DF. 0000.003D.0180.0800.0000.1130  ;3271  ERROR1    ;INTERRUPT ENABLE DIDN'T SET
U 120A. 0F00.003C.0180.0800.0000.120B  ;3272 SA0010: D_0
U 120B. 0000.003C.6D80.3C00.0000.120C  ;3273  ID[FAULT]_D   ;CLEAR FAULT INTERRUPT ENABLE
U 120C. 0F00.003C.8D80.0800.0084.720D  ;3274  SC_K[.1F],D_0
U 120D. 0801.001C.0180.0800.0000.120E  ;3275  D_D.ORNOT.MASK   ;GENERATE FORCE SBI P0 FAULT
U 120E. 0000.003C.75F0.2C00.0000.120F  ;3276  Q_ID[MAINT]
U 120F. 081D.0030.0180.0800.0000.1210  ;3277  D_D.OR.Q
U 1210. 0018.0038.1980.0800.0200.1211  ;3278  VA_K[ZERO]
U 1211. 0000.003C.7580.3C00.0000.1212  ;3279  ID[MAINT]_D   ;CACHE OFF, FORCE SBI P0 FAULT
U 1212. 0F00.003C.0180.0800.0000.1213  ;3280  D_0
U 1213. 0000.003C.0180.A800.0000.10E0  ;3281  MCT/WRITE.P
            ;3282 =0 D_K[.6],
U 10E0. 0818.0039.D580.0800.0000.1157  ;3283   CALL[WAIT.LOOP]
U 10E1. 0000.003C.0180.0800.0000.10E2  ;3284  NOP
U 10E2. 0000.003D.0180.0800.0000.112B  ;3285 =0 CALL[DISABLE.CACHE]  ; Disable cache and clear force
            ;3286       ; ...fault bits
U 10E3. 0000.003C.0180.0800.0000.10E4  ;3287  NOP
            ;3288
            ;3289 =0 D_K[.FF].LEFT,SI/MUL-,
U 10E4. 0838.0039.4B80.0800.0000.1157  ;3290   CALL[WAIT.LOOP]
U 10E5. 0000.003C.0180.0800.0000.1044  ;3291  NOP
U 1044. 0000.003D.0180.0800.0000.1060  ;3292 =00 CALL[NOINTR]
U 1045. 0000.003D.0180.0800.0000.1130  ;3293  ERROR1    ;IF AN SBI FAULT INTERRUPT
            ;3294       ; THE ENABLE IS STUCK ACTIVE
U 1046. 0000.003C.0180.0800.0000.10E6  ;3295  NOP
            ;3296 =
            ;3297 ;
            ;3298 ;================================================================================
            ;3299 ;*
            ;3300 ;SUBTEST  2
```

```
            ;3301 ;
            ;3302 ;   CHECK THE SILO INTERRUPT ENABLE BIT
            ;3303 ;   ID=IC,"COMP",BIT #30
            ;3304 ;
            ;3305 ; ERROR LOGOUT
            ;3306 ;
            ;3307 ; DATA: TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
            ;3308 ;  UNDEFINED
            ;3309 ;  UNDEFINED
            ;3310 ;  UNDEFINED
            ;3311 ;  UNDEFINED
            ;3312 ;  UNDEFINED
            ;3313 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
            ;3314 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
            ;3315 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
            ;3316 ;  INTERRUPT FLAG:
            ;3317 ;   0=>NO INTERRUPT OCCURRED
            ;3318 ;   1=>AN INTERRUPT OCCURRED
            ;3319 ;
            ;3320 ;-
            ;3321 ;;;
```

```
U 1049, 0000,003D,0180,0800,0000,1130  ;3356  ERROR1    ;IF INTERRUPT TAKEN IS FOR THE
          ;3357       ; SILO, THEN THE INTERRUPT
          ;3358       ; ENABLED IS STUCK ACTIVE
U 104A, 0000,003C,0180,0800,0000,1223  ;3359  NOP
U 1223, 0000,003C,0180,0800,0000,10F2  ;3360 = NOP
          ;3361 ;
          ;3362 ;================================================================================
          ;3363 ;+
          ;3364 ;SUBTEST  3
          ;3365 ;
          ;3366 ;   CHECK RDS/CRD INTERRUPT DISABLE
          ;3367 ;   ID=19, 'SBI.ERR",BIT #15
          ;3368 ;
          ;3369 ; ERROR LOGOUT
          ;3370 ;
          ;3371 ; DATA: TR LEVEL UNDER TEST; <15>=1=16K CTRLR FLAG
          ;3372 ;  UNDEFINED
          ;3373 ;  UNDEFINED
          ;3374 ;  UNDEFINED
          ;3375 ;  UNDEFINED
          ;3376 ;  UNDEFINED
          ;3377 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
          ;3378 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
          ;3379 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
          ;3380 ;  INTERRUPT FLAG:
          ;3381 ;   0=>NO INTERRUPT OCCURRED
          ;3382 ;   1=>AN INTERRUPT OCCURRED
          ;3383 ;
          ;3384 ;-
          ;3385 ;================================================================================
          ;3386 ;
U 10F2, 0000,003D,0180,0800,0000,1134  ;3387 =0 SUBTEST
U 10F3, 0000,003C,0180,0800,0000,10F4  ;3388  NOP
U 10F4, 0000,003D,0180,0800,0000,112F  ;3389 =0 ERLOOP
U 10F5, 0000,003C,0180,0800,0000,10F6  ;3390  NOP
U 10F6, 0000,003D,0180,0800,0000,1159  ;3391 =0 CALL[INIT]
U 10F7, 0F00,003C,0180,0800,0000,1224  ;3392  D_0
U 1224, 0000,003C,6580,3C00,0000,1225  ;3393  ID[SBI.ERR]_D
U 1225, 0000,003C,0180,0800,0000,1226  ;3394  NOP
U 1226, 0000,003C,65F0,2C00,0000,1227  ;3395  Q_ID[SBI.ERR]
U 1227, 0019,2034,4580,0800,0010,1228  ;3396  ALU_Q.AND.K[.8000],CLK.UBCC ;DID IT CLEAR?
U 1228, 0000,013C,0180,0800,0000,10F8  ;3397  Z?
U 10F8, 0000,003D,0180,0800,0000,1130  ;3398 =0 ERROR1    ;RDS,CRD INTERRUPT ENABLE
          ;3399       ;  DIDN'T CLEAR
U 10F9, 0818,0038,4580,0800,0000,1229  ;3400  D_K[.8000]
U 1229, 0000,003C,6580,3C00,0000,122A  ;3401  ID[SBI.ERR]_D   ;SEE IF INTERRUPT ENABLE WILL SET
U 122A, 0000,003C,65F0,2C00,0000,122B  ;3402  Q_ID[SBI.ERR]
U 122B, 0019,2034,4580,0800,0010,122C  ;3403  ALU_Q.AND.K[.8000],CLK.UBCC ;DID IT SET?
U 122C, 0000,013C,0180,0800,0000,10FA  ;3404  Z?
U 10FA, 0000,003C,0180,0800,0000,10FC  ;3405 =0 J/SB0010
U 10FB, 0000,003D,0180,0800,0000,1130  ;3406  ERROR1    ;RDS,CRD INTERRUPT ENABLE DID
          ;3407       ; NOT SET
          ;3408 =0
          ;3409 SB0010:
U 10FC, 0000,003D,0180,0800,0000,112F  ;3410  ERLOOP    ; Set the error looping mechanism
```

```
U 10FD. 0F00.003C.0180.0800.0000.122D  ;3411  D_0
U 122D. 0000.003C.6580.3C00.0000.122E  ;3412  ID[SBI.ERR]_D   ;CLEAR INTERRUPT ENABLE
U 122E. 0018.0038.1980.0800.0200.122F  ;3413  VA_K[ZERO]   ; Point to location 0
U 122F. 0818.0038.0580.0800.0000.1230  ;3414  D_K[.1]    ; Get data to force error in location 0
U 1230. 0000.003C.0180.A800.0000.1231  ;3415  CACHE.P_D[LONG]   ; Initialize location 0
U 1231. 0000.003C.0180.0803.0000.1232  ;3416  VA_VA+4    ; Point VA to locaiton 4
U 1232. 0818.0038.0D80.0800.0000.1233  ;3417  D_K[.3]    ; Get data to initialize locaiton 4
U 1233. 0000.003C.0180.A800.0000.10FE  ;3418  CACHE.P_D[LONG]   ; Initialize location 4
U 10FE. 0818.0039.0980.0800.0000.1175  ;3419 =0 SET.DIAG.MODE[.2]  ; Set Diagnostic Mode 2 (ECC substitute)
U 10FF. 0000.003C.0180.0800.0000.110A  ;3420  NOP
U 110A. 0818.0039.8580.0800.0000.116F  ;3421 =0 SET.ECC[.C]   ; Set ECC substitute bit to x0C
U 110B. 0000.003C.0180.0800.0000.1110  ;3422  NOP
U 1110. 0000.003D.8580.0800.0084.7153  ;3423 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 1111. 0018.0038.1980.0800.0200.1234  ;3424  VA_K[ZERO]   ; Point to location 0
U 1234. 0000.003C.0180.C800.0000.1112  ;3425  D[LONG]_CACHE.P   ; Read location forcing CRD error
          ;3426 =0 D_K[.6],
U 1112. 0818.0039.D580.0800.0000.1157  ;3427   CALL[WAIT.LOOP]  ; Wait a certain number of cycles
U 1113. 0000.003C.0180.0800.0000.1050  ;3428  NOP
U 1050. 0000.003D.0180.0800.0000.1060  ;3429 =00 CALL[NOINTR]   ; See if any interrupts were generated
U 1051. 0000.003D.0180.0800.0000.1130  ;3430  ERROR1    ; Unexpected Interrupt for CRD error
U 1052. 0000.003D.8580.0800.0084.7153  ;3431 =0 SET.TRAP.MASK[.C]  ; Enable RDS uTraps
U 1053. 0018.0038.1180.0800.0200.1235  ;3432  VA_K[.4]   ; Point VA to location 4
U 1235. 0000.003C.0180.C800.0000.1114  ;3433  D[LONG]_CACHE.P   ; Read location 4 forcing RDS error
          ;3434 =0 D_K[.6],
U 1114. 0818.0039.D580.0800.0000.1157  ;3435   CALL[WAIT.LOOP]  ; Wait a certain number of cycles
U 1115. 0000.003C.0180.0800.0000.1054  ;3436  NOP
U 1054. 0000.003D.0180.0800.0000.1060  ;3437 =00 CALL[NOINTR]   ; See if any interrupts were generated
U 1055. 0000.003D.0180.0800.0000.1130  ;3438  ERROR1    ; Unexpected Interrupt on RDS error
U 1056. 0000.003C.0180.0800.0000.1057  ;3439  NOP
U 1057. 0000.003C.0180.0800.0000.1116  ;3440 M10END: NOP    ; ALL DONE.
          ;3441
```

```
ZZ-ESKAR-V22  TEST 1D7 RESERVED                                                    SEQ 1572
ESKAR4E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:11              Page   9

            ;3442  .PAGE "TEST 1D7 RESERVED"
            ;3443 =0
U 1116. 0000,003D,0180,0800,0000,104F  ;3444 T1D7: NEWTST
U 1117. 0000,003C,0180,0800,0000,1118  ;3445  NOP
            ;3446
```

```
            ;3447  .PAGE "TEST 1D8 RESERVED"
            ;3448 =0
U 1118. 0000,003D,0180,0800,0000,104F  ;3449 T1D8: NEWTST
U 1119. 0000,003C,0180,0800,0000,1236  ;3450  NOP
            ;3451
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 1574
ESKAR4E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:11          Page   9

          ;3452  .PAGE "DUMMY NEWTST"
U 1236. 0000,003D,0180,0800,0000,104F  ;3453 DUM: NEWTST
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 2536= 2537= 2538= 2540=
U 1008  2647= 2648= 2649= 2650= 2651= 2652= 2653= 2654=
U 1010  2584= 2585= 2586= 2588= 1898= 1899= 1875 3335=
U 1018  2837= 2838= 2839= 1876 1958= 1960= 1877 3336=
U 1020  2844= 2845= 2846= 1878 2850= 2851= 2852= 1879
U 1028  2856= 2859= 2861= 2862= 3072= 3073= 3074= 1880
U 1030  3076= 3077= 3078= 1900 3123= 3124= 3125= 1901
U 1038  3127= 3128= 3129= 1902 3174= 3175= 3176= 1903
U 1040  3178= 3179= 3180= 1904 3292= 3293= 3295= 1905
U 1048  3355= 3356= 3359= 1906 1971= 1972= 2022: 1947
U 1050  3429= 3430= 3431= 3432= 3437= 3438= 3439= 3440=
U 1058  2015= 2016= 2047: 1948 2040= 2041= 2050: 1949
U 1060  2137= 2138= 2141= 2142= 2144= 2145= 2149= 2150=
U 1068  2183= 2184= 2270= 2271= 2286= 2287= 2289= 2290=
U 1070  2292= 2293= 2303= 2304= 2305= 2307= 2308= 2309=
U 1078  2310= 2311= 2312= 2313= 2314= 2315= 2316= 2317=
U 1080  2319= 2320= 2491= 2492= 2626= 2627= 2635= 2636=
U 1088  2637= 2639= 2642= 2643= 2684= 2685= 2686= 2687=
U 1090  2694= 2695= 2701= 2703= 2706= 2707= 2726= 2727=
U 1098  2729= 2730= 2896= 2897= 2900= 2901= 2912= 2913=
U 10A0  2916= 2917= 3026= 3027= 3028= 3029= 3049= 3050=
U 10A8  3051= 3052= 3053= 3054= 3058= 3059= 3070= 3071=
U 10B0  3081= 3082= 3085= 3086= 3100= 3101= 3102= 3103=
U 10B8  3104= 3105= 3109= 3110= 3121= 3122= 3132= 3133=
U 10C0  3136= 3137= 3151= 3152= 3153= 3154= 3155= 3156=
U 10C8  3160= 3161= 3172= 3173= 3183= 3185= 3187= 3188=
U 10D0  3219= 3220= 3221= 3222= 3250= 3251= 3252= 3253=
U 10D8  3254= 3255= 3257= 3258= 3264= 3266= 3270= 3271=
U 10E0  3283= 3284= 3285= 3287= 3290= 3291= 3323= 3324=
U 10E8  3325= 3326= 3327= 3328= 3330= 3331= 3345= 3346=
U 10F0  3353= 3354= 3387= 3388= 3389= 3390= 3391= 3392=
U 10F8  3398= 3400= 3405= 3406= 3410= 3411= 3419= 3420=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1950 3421= 3422= 1870: 1871: 1872: 1873:
U 1110  3423= 3424= 3427= 3428= 3435= 3436= 3444= 3445=
U 1118  3449= 3450= 1951 1952 1953 1954 1955 1956
U 1120  1957 1961 1962 1963 1964 1965 1966 1967
U 1128  1968 1969 1970 1991 1992 1993 1994 1996
U 1130  2020 2021 2045 2046 2078 2079 2086 2087
U 1138  2093 2094 2095 2109 2110 2111 2112 2113
U 1140  2139 2140 2146 2147 2148 2151 2152 2153
U 1148  2154 2155 2156 2157 2158 2160 2185 2186
U 1150  2187 2188 2211 2238 2239 2070: 2240 2265
U 1158  2284 2302 2321 2322 2351 2352 2353 2361
U 1160  2362 2363 2364 2372 2373 2374 2375 2383
U 1168  2384 2385 2386 2393 2394 2395 2396 2420
U 1170  2421 2422 2423 2424 2425 2449 2450 2451
U 1178  2452 2453 2454 2480 2481 2482 2483 2484
U 1180  2485 2486 2487 2489 2490 2628 2629 2630
U 1188  2631 2640 2641 2645 2646 2659 2660 2662
U 1190  2663 2664 2666 2671 2672 2673 2675 2680
U 1198  2682 2683 2691 2692 2696 2697 2698 2699
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1576
ESKAR4E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:11     Page   9

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2700 2708 2709 2710 2712 2713 2714 2715
U 11A8  2716 2717 2722 2724 2725 2728 2753 2754
U 11B0  2755 2756 2785 2786 2788 2789 2790 2835
U 11B8  2836 2863 2892 2894 2895 2899 2903 2904
U 11C0  2905 2906 2908 2909 2910 2911 2914 2915
U 11C8  3030 3032 3033 3034 3055 3056 3057 3060
U 11D0  3061 3062 3063 3064 3065 3066 3067 3068
U 11D8  3075 3079 3080 3083 3084 3099 3106 3107
U 11E0  3108 3111 3112 3113 3114 3115 3116 3117
U 11E8  3118 3119 3126 3130 3131 3134 3135 3150
U 11F0  3157 3158 3159 3162 3163 3164 3165 3166
U 11F8  3167 3168 3169 3170 3177 3181 3182 3186
U 1200  3189 3256 3259 3260 3261 3262 3263 3267
U 1208  3268 3269 3272 3273 3274 3275 3276 3277
U 1210  3278 3279 3280 3281 3329 3332 3333 3334
U 1218  3338 3339 3340 3341 3342 3343 3344 3348
U 1220  3349 3350 3351 3360 3393 3394 3395 3396
U 1228  3397 3401 3402 3403 3404 3412 3413 3414
U 1230  3415 3416 3417 3418 3425 3433 3453
U 1238  - 12A7 Unused
U 12A8    2071:
U 12B0  - 13EF Unused
U 13F0  2054: 2055: 2056: 2057: 2058: 2059: 2060: 2061:
U 13F8  2062: 2063: 2064: 2065: 2066: 2067: 2068: 2069:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1577
ESKAR4E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:11     Page   9

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR4E        584

Used               584
Remaining

Total microwords used in memory U: 584
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1578
ESKAR4E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:11     Page   9


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4E.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 1579
ESKAR4F.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1918 Micro Monitor Interface Routines
; 1919   Newtest Routine
; 1976   Error Loop Routine
; 1993   Error 1 Routine
; 2018   ERROR1 WITH PARAMETER
; 2039   Error 2 Routine
; 2065   ERROR 2 WITH PARAMETER
; 2084   Micro-Monitor Call
; 2145   RESET SUBTEST NUMBER
; 2168 DIAGNOSTIC SPECIFIC ROUTINES
; 2169   Disable Cache Routine
; 2191   FIND MS780-E MEMORY
; 2333   Generate IO Address
; 2359   Set Starting Address
; 2393   ENABLE NEXT MS780-E
; 2445   Select Controller
; 2483   SELECT LOWER CONTROLLER
; 2531   SELECT UPPER CONTROLLER
; 2579   Set Trap Mask
; 2608   CONFIGURE MS780-E/H
; 2682   ENBFLT ROUTINE
; 2699   NOINTR ROUTINE
; 2785   Wait Loop Routine
; 2816   Initialize the SBI
; 2866   INIT ROUTINE
; 2952 TEST 1D9 CANCEL TEST FOR SBI CYCLES
; 3772 TEST 1DA RESERVED
; 3777 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808   ;*****************************************************************************
;1809   ;
;1810   ;
;1811   ; MODULE NAME: ESKAR4F
;1812   ;
;1813   ; CREATION DATE:  SEPTEMBER 1982
;1814   ; AUTHOR:  DAN GRIGORE
;1815   ;
;1816   ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817   ;
;1818   ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819   ;
;1820   ; -  WHAT CHANGE WAS MADE
;1821   ;
;1822   ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823   ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824   ;
;1825   ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826   ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827   ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828   ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829   ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830   ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831   ;    ESKAR3C.
;1832   ;
;1833   ; START OF REVISION HISTORY:
;1834   ;
;1835   ;
;1836   ;
;1837   ;
;1838   ;
;1839   ;*****************************************************************************
;1840
```

```
            ;1841  .PAGE
            ;1842
            ;1843  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7001  ;1862 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7001  ;1863 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7001  ;1864 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7001  ;1865 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7001  ;1866 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7001  ;1867 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7001  ;1868 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7001  ;1869 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7001  ;1870 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D, 0000.003C.8980.0800.0084.7001  ;1871 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7001  ;1872 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.1017  ;1874 TRPH0: Q_ID[USTACK]
U 1017, 0C00.003C.01E0.0800.0000.101B  ;1875  Q_D,D_Q
U 101B, 0000.003C.8180.3C00.0000.101F  ;1876  ID[USTACK]_D
U 101F, 0C00.003C.01E0.0800.0000.1027  ;1877  Q_D,D_Q
U 1027, 0000.003C.01C0.0A78.0000.102B  ;1878  Q_R[0F]
U 102B, 0001.2024.0180.0800.0010.102F  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102F, 0000.013C.0180.0800.0000.1002  ;1880  Z?    ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.104C  ;1897 TRPH1: RC[0D]_SC
U 104C, 0000.003D.D980.0800.0084.71DB  ;1898 =0 SETRCF[.9]
U 104D, 0000.003C.49F0.2C00.0000.1033  ;1899  Q_ID[TBER0]
U 1033, 0001.203C.4DF0.2DB0.0000.1037  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1037, 0001.203C.65F0.2DB8.0000.103B  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 103B, 0001.203C.6DF0.2DD8.0000.103F  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 103F, 0001.203C.75F0.2DE0.0000.1043  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1043, 0001.203C.79F0.2DC8.0000.1047  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1047, 0001.203C.69F0.2DC0.0000.104B  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 104B, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.10AB  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
            ;1916
```

```
            ;1917  .PAGE
            ;1918  .TOC " Micro Monitor Interface Routines"
            ;1919  .TOC "   Newtest Routine"
            ;1920 ;++
            ;1921 ; FUNCTIONAL DESCRIPTION:
            ;1922 ;
            ;1923 ; This routine Initializes the following registers:
            ;1924 ;  PSL    to 1F
            ;1925 ;  CES    to 0
            ;1926 ;  ACC.CS    to disable accellerator
            ;1927 ;  SIR    to 0
            ;1928 ;  CLK.CS    to stop interval timer
            ;1929 ;  TBER0    to disable the TB
            ;1930 ;  SBI.MAINT to    0
            ;1931 ;  SBI.ERR    to 0
            ;1932 ;  FAULT    to 0
            ;1933 ;  COMP    to 0
            ;1934 ;  RC[0E]    to  0
            ;1935 ;  R[0F]    to 0
            ;1936 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1937 ; A SCOPE call is then made to the Monitor.
            ;1938 ;
            ;1939 ; CALLING CONSTRAINT:
            ;1940 ;
            ;1941 ; =0
            ;1942 ;
            ;1943 ; SIDE EFFECTS:
            ;1944 ;
            ;1945 ; D Reg is destroyed
            ;1946 ; SC is destroyed
            ;1947 ;--
U 104F. 0000.003C.65F8.0800.0084.7053  ;1948 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1053. 0818.0038.8D80.0800.0000.1057  ;1949  d_k[.1f]  ; D=1F
U 1057. 0D00.003C.0180.0800.0000.105B  ;1950  d_dal.sc  ; D=001F0000
U 105B. 0000.003C.3D80.3C00.0000.105F  ;1951  id[psl]_d  ; psl = 001F0000
U 105F. 0818.0038.0580.0800.0000.1063  ;1952  d_k[.1]  ; Clear the CES register
U 1063. 0D00.003C.0180.0800.0000.1067  ;1953  d_dal.sc  ; D = 00010000
U 1067. 0F00.003C.3180.3C00.0000.106B  ;1954  id[ces]_d,d_0  ; ces = 00010000
U 106B. 0000.003C.5D80.3C00.0000.106F  ;1955  id[acc.cs]_d  ; acc.cs = 0
U 106F. 0000.003C.3980.3C00.0000.1073  ;1956  id[sir]_d  ; sir = 0
U 1073. 0000.003C.2980.3C00.0000.107B  ;1957  id[clk.cs]_d  ; clk.cs = 0
U 107B. 0000.003C.4980.3C00.0000.1058  ;1958  id[tber0]_d  ; tber0 = 0
U 1058. 0000.003D.0180.0800.0000.1260  ;1959 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1960  intrpt.strobe,  ; Strobe interrupts
U 1059. 0818.0038.C180.0800.4000.107F  ;1961      d_k[.ffff]  ; Setup to clear SBI error register and
U 107F. 0801.0028.6580.3C00.0000.1083  ;1962  id[sbi.err]_d,d_not.d ; and fault register
U 1083. 0F00.003C.6D80.3C00.0000.1087  ;1963  id[fault]_d,d_0  ; ...
U 1087. 0000.003C.6D80.3C00.0000.108B  ;1964  id[fault]_d  ; fault = 0
U 108B. 0000.003C.7580.3C00.0000.108F  ;1965  id[maint]_d  ; MAINT = 0
U 108F. 0000.003C.6580.3C00.0000.1093  ;1966  id[sbi.err]_d  ; sbi error reg = 0
U 1093. 0000.003C.7180.3C00.0000.1097  ;1967  id[comp]_d  ; comp reg = 0
U 1097. 0000.003C.E580.3C00.0000.109B  ;1968  id[T9]_d  ; Clear code for subroutine START.TEST
U 109B. 0001.003C.0180.09F0.0000.109F  ;1969  rc[0e]_d  ;
U 109F. 0001.003C.0180.0AF8.0000.10A3  ;1970  r[0f]_d  ; Clear expected trap mask
U 10A3. 0001.003C.0180.09F8.0000.105C  ;1971  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 105C, 0000,003D,0180,0800,0000,11DF  ;1972 =0 call[disable.cache] ; Disable the cache
U 105D, 0F03,003C,0180,09E8,0000,105E  ;1973  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
            ;1974
```

```
            ;1975  .PAGE
            ;1976  .TOC "   Error Loop Routine"
            ;1977 ;++
            ;1978 ; FUNCTIONAL DESCRIPTION:
            ;1979 ;
            ;1980 ; This routine is called with the "ERLOOP" macro. The
            ;1981 ; routine calls the Micro Monitor to setup an error loop.
            ;1982 ;
            ;1983 ; CALLING CONSTRAINT:
            ;1984 ;
            ;1985 ; =0
            ;1986 ;
            ;1987 ; SIDE EFFECTS:
            ;1988 ;
            ;1989 ; D Reg - Destroyed
            ;1990 ;--
U 10A7, 0818,0038,0980,0800,0000,105E  ;1991 ERLOOP: d_k[.2],j/callcon
```

```
            ;1992  .PAGE
            ;1993  .TOC "   Error 1 Routine"
            ;1994 ;**
            ;1995 ; FUNCTIONAL DESCRIPTION:
            ;1996 ;
            ;1997 ; This routine is used to report an error to the user. This
            ;1998 ; routine is used when you do not wish to continue in the
            ;1999 ; same test after the call to the Monitor.
            ;2000 ;
            ;2001 ; CALLING CONSTRAINT:
            ;2002 ;
            ;2003 ; None
            ;2004 ;
            ;2005 ; INPUTS:
            ;2006 ;
            ;2007 ; rc[0f]<15:8> - Contain the subtest number
            ;2008 ;
            ;2009 ; OUTPUTS:
            ;2010 ;
            ;2011 ; D<15:8> - Subtest number
            ;2012 ; D<7:0> - Error 1 Monitor Code
            ;2013 ;--
U 10AB, 0810,0038,0180,0978,0000,10AF  ;2014 ERROR1: d_rc[0f]  ; Get the subtest number
U 10AF, 0819,0034,4D80,0800,0000,104E  ;2015  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2016 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2017  .PAGE
            ;2018  .TOC "   ERROR1 WITH PARAMETER"
            ;2019 ;**
            ;2020 ; FUNCTIONAL DESCRIPTION:
            ;2021 ;
            ;2022 ; This subroutine takes  as parameter the number of arguments
            ;2023 ; to be printed to the console in the case of an error logout.
            ;2024 ;
            ;2025 ; CALLING CONSTRAINT:
            ;2026 ;
            ;2027 ; =0
            ;2028 ;INPUTS:
            ;2029 ;
            ;2030 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2031 ;--
            ;2032 =0
            ;2033 ERR1.ARG:
U 10BC, 0000,003D,0180,0800,0000,11DB  ;2034  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10BD, 0000,003C,0180,0800,0000,10AB  ;2035  J/ERROR1  ; Go to ERROR1
            ;2036 ;
            ;2037 ;
```

```
            ;2038  .PAGE
            ;2039  .TOC "   Error 2 Routine"
            ;2040 ;**
            ;2041 ; FUNCTIONAL DESCRIPTION:
            ;2042 ;
            ;2043 ; This routine is used  to report an error to the user. This
            ;2044 ; routine is used when you wish to continue in the same test
            ;2045 ;       after the call to the Monitor.
            ;2046 ;
            ;2047 ; CALLING CONSTRAINT:
            ;2048 ;
            ;2049 ; =0
            ;2050 ;
            ;2051 ; INPUTS:
            ;2052 ;
            ;2053 ; rc[0f]<15:8> - Contain the subtest number
            ;2054 ;
            ;2055 ; OUTPUTS:
            ;2056 ;
            ;2057 ; D<15:8> - Subtest number
            ;2058 ; D<7:0> - Error 2 Monitor Code
            ;2059 ;--
U 10B3,  0810,0038,0180,0978,0000,10B7  ;2060 ERROR2: d_rc[0f]  ; Get the subtest number
U 10B7,  0819,0034,4D80,0800,0000,105A  ;2061  d_d.and.k[.ff00] ; ...
U 105A,  0819,0030,D580,0800,0000,105E  ;2062 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2063 ;
```

```
            ;2064  .PAGE
            ;2065  .TOC "    ERROR 2 WITH PARAMETER"
            ;2066 ;++
            ;2067 ; FUNCTIONAL DESCRIPTION:
            ;2068 ;
            ;2069 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2070 ; except that in this case after setting the number of arguments
            ;2071 ; too the console, control is transferred to routine ERROR2.
            ;2072 ;
            ;2073 ; INPUTS:
            ;2074 ;
            ;2075 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2076 ;
            ;2077 ;--
            ;2078 =0
            ;2079 ERR2.ARG:
U 10BE, 0000,003D,0180,0800,0000,11DB  ;2080  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 10BF, 0000,003C,0180,0800,0000,10B3  ;2081  J/ERROR2  ; Go to ERROR2
            ;2082 ;
```

```
            ;2083  .PAGE
            ;2084  .TOC "   Micro-Monitor Call"
            ;2085 ;*
            ;2086 ; FUNCTIONAL DESCRIPTION:
            ;2087 ;
            ;2088 ; This micro word calls the micro monitor.
            ;2089 ;
            ;2090 ; EXPLICIT INPUTS:
            ;2091 ;
            ;2092 ; D reg must contain valid monitor code.
            ;2093 ;--
            ;2094 105E:
            ;2095 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2096  id[txdb]_d,j/callcon ; Send code to monitor and hang
            ;2097
            ;2098 ;*
            ;2099 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2100 ;-
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2101 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2102 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2103 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2104 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2105 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2106 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2107 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2108 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2109 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2110 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2111 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2112 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2113 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2114 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2115 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;2116 13FF: NOP
U 1155, 0000,003C,0180,0800,0000,12AA  ;2117 1155: NOP
U 12AA, 0000,003C,0180,0800,0000,10BB  ;2118 12AA: NOP
            ;2119 ;*
            ;2120 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2121 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2122 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2123 ; TYPING IT.
            ;2124 ;-
U 10BB, 0810,0038,4D80,0978,0000,1109  ;2125 SUBTST: D_RC[0F],KMX/.FF00
U 1109, 0019,4016,4D80,09F8,0000,0001  ;2126  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2127 ;*****************************************************************
            ;2128 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2129 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2130 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2131 ;-
            ;2132 SUBTEST1:
U 1154, 0810,0038,0180,0978,0000,11DA  ;2133  D_RC[0F]
U 11DA, 0019,0032,4D80,09F8,0000,0001  ;2134  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2135 ;*
            ;2136 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2137 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
```

```
            :2138 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            :2139 ;-
U 11DB, 0010,0038,01C0,0978,0000,11DC  ;2140 SETRCF: Q_RC[0F]
U 11DC, 0019,2034,4DC0,0800,0000,11DD  ;2141  Q_Q.AND.K[.FF00]
U 11DD, 0019,2032,1D80,09F8,0000,0001  ;2142  RC[0F]_Q.OR.SC,RETURN1
            :2143
```

```
            ;2144  .PAGE
            ;2145  .TOC "    RESET SUBTEST NUMBER"
            ;2146 ;**
            ;2147 ; FUNCTIONAL DESCRIPTION:
            ;2148 ;
            ;2149 ; Since some  of the tests will have to be restarted when two
            ;2150 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2151 ; the subtest counter to zero ( only for thoes tests that have
            ;2152 ; more than one subtest). This subroutine may also be necessary
            ;2153 ; when a test is being loop on.
            ;2154 ;
            ;2155 ; CALLING CONSTRAINT:
            ;2156 ;
            ;2157 ; =0
            ;2158 ; SIDE EFFECTS:
            ;2159 ;
            ;2160 ; D is destroyed
            ;2161 ;
            ;2162 ;--
            ;2163 RESET.SUBTST:
            ;2164  RC[0F] 0,   ; Reset subtest number
U 11DE, 0003,003E,0180,09F8,0000,0001  ;2165   RETURN1   ; ...and return
            ;2166
```

```
            ;2167  .PAGE
            ;2168  .TOC " DIAGNOSTIC SPECIFIC ROUTINES"
            ;2169  .TOC "   Disable Cache Routine"
            ;2170 ;++
            ;2171 ; FUNCTIONAL DESCRIPTION:
            ;2172 ;
            ;2173 ; This routine disables cache hits by setting bits <16:15>
            ;2174 ; in the maintenance register.
            ;2175 ;
            ;2176 ; CALLING SEQUENCE:
            ;2177 ;
            ;2178 ; =0
            ;2179 ;
            ;2180 ; SIDE EFFECTS:
            ;2181 ;
            ;2182 ; D & Q Registers destroyed
            ;2183 ;--
            ;2184 DISABLE.CACHE:
U 11DF, 0818,0038,4580,0800,0000,11E0  ;2185  d_k[.8000]  ; ... and seed constant
U 11E0, 0500,003C,0180,0800,0000,11E1  ;2186  d_d.left  ; Generate final constant
U 11E1, 0819,0030,4580,0800,0000,11E2  ;2187  d_d.or.k[.8000]  ; ... = 00018000
U 11E2, 0000,003E,7580,3C00,0000,0001  ;2188  id[maint]_d,return1 ; Disable cache and return
            ;2189
```

```
                ;2190  .PAGE
                ;2191  .TOC "   FIND MS780-E MEMORY"
                ;2192 ;++
                ;2193 ; FUNCTIONAL DESCRIPTION:
                ;2194 ;
                ;2195 ;          The diagnostic is designed to handle up to 4 (four)
                ;2196 ;     MS780-E/H's on the 780 SBI. This routine looks at all TR levels
                ;2197 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
                ;2198 ;     3D will hold the I/O base address of the MS780-E subsystems found
                ;2199 ;     on the SBI, and ID Register 3E will hold the Total number of
                ;2200 ;     MS780-E/H's found minus one. Also, it is to be noted that the
                ;2201 ;     first MS780-E found will have its starting address set to 0
                ;2202 ;     (zero).
                ;2203 ;     All the other MS780-E/H's found will have their starting address
                ;2204 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
                ;2205 ;     Reg 3E to decide if there are any more MS780-E and will take
                ;2206 ;     apropriate action.  Register RC[0E] is used as a pointer to the
                ;2207 ;     current MS780-E unit under test.
                ;2208 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
                ;2209 ;     set to maximum.
                ;2210 ;
                ;2211 ; CALLING CONSTRAINT:
                ;2212 ;
                ;2213 ; =00
                ;2214 ;
                ;2215 ; OUTPUTS:
                ;2216 ;
                ;2217 ; rc[0e] - Contains address of Configuration Register
                ;2218 ; r[0]  - Contains TR level
                ;2219 ;
                ;2220 ; SIDE EFFECTS:
                ;2221 ;
                ;2222 ; D register is destroyed.
                ;2223 ;--
                ;2224 =0
                ;2225 FIND.MS780E:
U 10C0, 0000,003D,0180,0800,0000,10EA  ;2226  call[sbi.init]  ; Make sure SBI is enabled
U 10C1, 0003,003C,0180,0988,0000,11E3  ;2227  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11E3, 0818,0038,1980,0800,0000,11E4  ;2228  d_k[zero]  ; Clear MS780-E/H counter
U 11E4, 0000,003C,F980,3C00,0000,11E5  ;2229  id[3e]_d  ; ...
U 11E5, 0018,0038,0580,0A80,0000,11E6  ;2230  r[0]_k[.1]  ; Init TR counter
U 11E6, 0F03,003C,0180,09F0,0000,10C2  ;2231  d_0,rc[0E]_0  ; Clear "Save" location for
                ;2232      ; ...CNFG A Reg
                ;2233 =0
                ;2234 FIND.MEM.LOOP:
U 10C2, 0800,003D,0180,0A00,0000,120C  ;2235  d_r[0],call[generate.io]; Generate address of TR level
U 10C3, 0001,003C,0180,09F0,0200,10C4  ;2236  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10C4, 0000,003D,8980,0800,0084,722D  ;2237 =0 set.trap.mask[.d] ; Enable timeout micro traps
                ;2238  d[long]_cache.p, ; Read the specified tr level
U 10C5, 0000,003C,0180,C970,0000,11E7  ;2239   ic_rc[0e]  ; ...
U 11E7, 0000,003C,0180,0A78,0010,11E8  ;2240  alu_r[0f],clk.ubcc ; Check for no response
U 11E8, 0001,013C,0180,0880,0082,10C6  ;2241  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 10C6, 0000,003C,0180,0800,0000,11E9  ;2242 =0 j/check.adpt.code ; Check for a memory
U 10C7, 0000,003C,0180,0800,0000,1208  ;2243  j/find.mem.lp1  ; Try next tr
                ;2244 CHECK.ADPT.CODE:
```

```
U 11E9, 0000,003C,1D80,0800,1404,71EA  ;2245  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11EA, 0000,163C,0180,0800,0000,1008  ;2246  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,71EB  ;2247 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,71EC  ;2248  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,1208  ;2249  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,1208  ;2250  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,71F1  ;2251  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,1208  ;2252  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,11F5  ;2253  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,11F5  ;2254  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2255   ;
            ;2256   ; Found an MS780-A or -C. Set the starting address
            ;2257   ; to maximum.
            ;2258   ;
U 11EB, 0818,0038,5D80,0800,0000,11ED  ;2259 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11EC, 0818,0038,0580,0800,0000,11ED  ;2260 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2261 MS780.COMMON:
U 11ED, 0819,0030,7180,0800,0000,11EE  ;2262  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11EE, 0000,003C,01F8,0803,0080,D1EF  ;2263  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11EF, 0D00,003C,0180,0800,0000,11F0  ;2264  d_dal.sc  ; Put data in bits<29:14>
            ;2265  cache.p_d[long], ; Set the starting address to maximum
U 11F0, 0000,003C,0180,A800,0000,1208  ;2266       j/find.mem.lp1 ; and continue TR scan
            ;2267   ;
            ;2268   ; Found an MA780. Set the starting address to maximum
            ;2269   ;
            ;2270 MA780.MAX:
U 11F1, 0818,0038,39F8,0803,0000,11F2  ;2271  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11F2, 0D00,003C,0180,0803,0000,11F3  ;2272  d_dal.sc,va_va+4 ; Put in proper position
U 11F3, 0000,003C,0180,0803,0000,11F4  ;2273  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2274  cache.p_d[long], ; Set starting address to maximum
U 11F4, 0000,003C,0180,A800,0000,1208  ;2275       j/find.mem.lp1
            ;2276 ;
            ;2277 ; Found an MS780E
            ;2278 ;
            ;2279 FOUND.MS780E:
U 11F5, 0000,003C,F9F0,2C00,0000,11F6  ;2280  q_id[3e]  ; Set up to see how many MS780-E's
            ;2281  alu_q.xor.k[.3], ; Are there Maximum units?
U 11F6, 0019,2020,0D80,0800,0010,11F7  ;2282   clk.ubcc ; Check condition codes
U 11F7, 0000,013C,0180,0800,0000,10C8  ;2283  z?   ; Are we at maximum units for system
U 10C8, 0000,003C,0180,0800,0000,11F8  ;2284 =0 J/ms780e.tst  ; No go find how many units ther are
U 10C9, 0818,0038,C180,0800,0000,10CA  ;2285  d_k[.ffff]  ; Yes set address to maximum
U 10CA, 0000,003D,0180,0800,0000,1210  ;2286 =0 call[set.start.addr] ; .....
U 10CB, 0000,003C,0180,0800,0000,1208  ;2287  j/find.mem.lp1  ; Go check to see if we are done
            ;2288 ;
            ;2289 MS780E.TST:
            ;2290  alu_rc[01],  ; Check "Found MS780E Flag"
U 11F8, 0010,0038,0180,0908,0010,11F9  ;2291   clk.ubcc ; Check condition codes
U 11F9, 0810,0138,0180,0970,0000,10CC  ;2292  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2293 =0 j/not.first.ms780e, ; No
U 10CC, 0818,0038,C180,0800,0000,10D0  ;2294   d_k[.ffff] ; Set starting address
U 10CD, 0001,003C,0180,0988,0000,11FA  ;2295  rc[01]_d  ; Yes put base address in rc[01]
U 11FA, 0818,0038,7980,0800,0000,11FB  ;2296  d_k[.30]  ; Set up SC to point to first unit
U 11FB, 0019,0014,F580,0800,0082,11FC  ;2297  alu_d+k[.a],sc_alu ; ...
U 11FC, 0810,0038,0180,0908,0000,11FD  ;2298  d_rc[01]  ; Put base address of unit in D
U 11FD, 0000,003C,0180,3400,0000,11FE  ;2299  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11FE, 0F00,003C,0180,0800,0000,10CE  ;2300  d_0   ; Prepare to set starting address to Zero
U 10CE, 0000,003D,0180,0800,0000,1210  ;2301 =0 call[set.start.addr] ; Go do it
          ;2302  j/find.mem.lp1,  ; Check to see if we are done
U 10CF, 0003,003C,0180,09E8,0000,1208  ;2303   rc[0d]_0 ; ....
          ;2304 =0
          ;2305 NOT.FIRST.MS780E:
U 10D0, 0000,003D,0180,0800,0000,1210  ;2306  call[set.start.addr] ; Go set starting address to maximum
U 10D1, 0000,003C,F9F0,2C00,0000,11FF  ;2307  q_id[3e]  ; Get the number of MS780-Es found
U 11FF, 0819,2014,0580,0800,0000,1200  ;2308  d_q+k[.1]  ; Increment this counter
U 1200, 0000,003C,F980,3C00,0000,1201  ;2309  id[3e]_d  ; Save the counter
U 1201, 0018,0038,11C0,0800,0000,1202  ;2310  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2311     ; ...were the I/O base addresses will be stored
U 1202, 081D,2000,7980,0800,0000,1203  ;2312  d_q-d,k[.30]  ; ... adjust pointer
U 1203, 0819,0014,7980,0800,0000,1204  ;2313  d_d+k[.30]  ; Point the SC to the ID register
U 1204, 0000,003C,F580,0800,0000,1205  ;2314  k[.a]   ; ...to receive I/O base address
U 1205, 0019,0014,F580,0800,0082,1206  ;2315  alu_d+k[.a],sc_alu ; ...
U 1206, 0810,0038,0180,0970,0000,1207  ;2316  d_rc[0e]  ; Get base address to d
U 1207, 0000,003C,0180,3400,0000,1208  ;2317  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2318   ;
          ;2319   ; Try next tr
          ;2320   ;
          ;2321 FIND.MEM.LP1:
U 1208, 0818,0014,0580,0A80,0000,1209  ;2322  r[0]_la+k[.1],d_alu ; Increment TR level
          ;2323  alu_d.and.k[.f],
U 1209, 0019,0034,6180,0800,0010,120A  ;2324      clk.ubcc  ; Check if all done
U 120A, 0000,013C,0180,0800,0000,10D2  ;2325  z?   ; Branch if yes
U 10D2, 0000,003C,0180,0800,0000,10C2  ;2326 =0 j/find.mem.loop  ; Try next TR
U 10D3, 0810,0038,0180,0908,0010,120B  ;2327  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 120B, 0000,013C,0180,0800,0000,10D4  ;2328  z?   ; Check condition codes
U 10D4, 0001,003E,0180,09F0,0000,0002  ;2329 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10D5, 0000,003E,0180,0800,0000,0001  ;2330  return1   ; No MS780E found
          ;2331
```

```
            ;2332  .PAGE
            ;2333  .TOC "   Generate IO Address"
            ;2334 ;++
            ;2335 ; FUNCTIONAL DESCRIPTION:
            ;2336 ;
            ;2337 ; This routine is used to generate an SBI Configuration Register
            ;2338 ; address from a TR level.
            ;2339 ;
            ;2340 ; CALLING CONSTRAINT:
            ;2341 ;
            ;2342 ; =0
            ;2343 ;
            ;2344 ; INPUTS:
            ;2345 ;
            ;2346 ; D - Contains TR level
            ;2347 ;
            ;2348 ; OUTPUTS:
            ;2349 ;
            ;2350 ; D - Contains SBI IO address
            ;2351 ;--
            ;2352 GENERATE.IO:
U 120C, 0000,003C,89F8,0800,0084,720D  ;2353  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 120D, 0D00,003C,8D80,0800,0084,720E  ;2354  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 120E, 0000,003C,0980,0800,0084,B20F  ;2355  sc_sc-k[.2]  ; insert bit 29
U 120F, 0801,001E,0180,0800,0000,0001  ;2356  d_d.ornot.mask,return1 ; and return
            ;2357
```

```
            ;2358  .PAGE
            ;2359  .TOC "   Set Starting Address"
            ;2360 ;++
            ;2361 ; FUNCTIONAL DESCRIPTION:
            ;2362 ;
            ;2363 ; This routine is used to set the starting address of the
            ;2364 ; memory to the specified value.
            ;2365 ;
            ;2366 ; CALLING CONSTRAINT:
            ;2367 ;
            ;2368 ; =0
            ;2369 ;
            ;2370 ; INPUTS:
            ;2371 ;
            ;2372 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2373 ;     (B Register Image divided by 2**16)
            ;2374 ; rc[0] - Contains Address of Register A
            ;2375 ;
            ;2376 ; OUTPUTS:
            ;2377 ;
            ;2378 ; d - Contains aligned starting address (B Register Image)
            ;2379 ;
            ;2380 ; SIDE EFFECTS:
            ;2381 ;
            ;2382 ; d,q,va, and sc are destroyed
            ;2383 ;--
            ;2384 SET.START.ADDR:
U 1210, 0010,0038,6580,0970,0284,7211  ;2385  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1211, 0000,003C,01F8,0803,0000,1212  ;2386  va_va+4,q_0  ; Set address to Register B
            ;2387  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1212, 0D00,003C,0980,0800,0084,B213  ;2388       sc_sc-k[.2]  ; Setup to insert bit 14
U 1213, 0801,001C,0180,0800,0000,1214  ;2389  d_d.ornot.mask  ; Insert Write Enable bit
U 1214, 0000,003E,0180,A800,0000,0001  ;2390  cache.p_d[long],return1 ; Set the starting address and return
            ;2391
```

```
            ;2392  .PAGE
            ;2393  .TOC "   ENABLE NEXT MS780-E"
            ;2394 ;++
            ;2395 ; FUNCTIONAL DESCRIPTION:
            ;2396 ;
            ;2397 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2398 ;     They will be tested  separately, according to the TR level at which
            ;2399 ;     they are located,  starting with the one  at the lowest level first.
            ;2400 ;          When  a test has finished  with the first unit, it will have to call
            ;2401 ;     this specific  routine  to see  if any other MS780-E unit is present
            ;2402 ;     on the SBI. If more units are  present, the first one  will be  dis-
            ;2403 ;     abled  by  setting  its  starting  address to maximum, the next unit
            ;2404 ;     will be  enabled  by  setting its starting address to 0 and the test
            ;2405 ;     will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2406 ;     for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2407 ;     ID registers 3A through 3D and  a count of the Total number of units
            ;2408 ;     found - 1 in register 3E. In this subroutine the SC register is used
            ;2409 ;     as  a  pointer  to  ID  registers 3A through 3D.
            ;2410 ;
            ;2411 ; CALLING CONSTRAINT:
            ;2412 ;
            ;2413 ;  =00
            ;2414 ;
            ;2415 ;--
            ;2416 ENABLE.NEXT.MS780E:
U 1215, 0000,003C,F9F0,2C00,0000,1216  ;2417  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2418  ALU Q,   ; Check to see if it is zero
U 1216, 0001,203C,0180,0800,0010,1217  ;2419     CLK.UBCC  ; Check Condition Codes
U 1217, 0000,013C,0180,0800,0000,10D6  ;2420  Z?   ; ...for zero
U 10D6, 0000,003C,0180,0800,0000,1218  ;2421 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10D7, 0000,003E,0180,0800,0000,0002  ;2422  RETURN2   ; Zero No more units to test
            ;2423 ENABLE.NEXT:
U 1218, 0818,0038,C180,0800,0000,10D8  ;2424  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10D8, 0000,003D,0180,0800,0000,1210  ;2425 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10D9, 0818,0038,7980,0800,0000,1219  ;2426  D_K[.30]  ; Prepare to point to the ID register holding
            ;2427     ; ...the I/O Base address of the next MS780-E
U 1219, 0819,0014,1180,0800,0000,121A  ;2428  D_D+K[.4]  ; ...
U 121A, 0000,003C,F580,0800,0000,121B  ;2429  K[.A]   ; ...
U 121B, 0819,0014,F580,0800,0000,121C  ;2430  D_D+K[.A]  ; ...
U 121C, 0000,003C,F9F0,2C00,0000,121D  ;2431  Q_ID[3E]  ; Get MS780-E Counter
            ;2432  D_D-Q,   ; D now holds the pointer to the ID register
U 121D, 081D,0000,0180,0800,0082,121E  ;2433    SC_ALU  ; ...
U 121E, 0000,003C,01F0,2400,0000,121F  ;2434  Q_ID(SC)  ; Q gets adrress of next MS780E
U 121F, 0001,203C,0180,09F0,0000,1220  ;2435  RC[0E]_Q  ; Save it also in RC[0E]
U 1220, 0F03,003C,0180,09E8,0000,10DA  ;2436  RC[0D]_0,D_0  ; Set starting address to zero
U 10DA, 0000,003D,0180,0800,0000,1210  ;2437 =0 CALL[SET.START.ADDR] ; ....
U 10DB, 0F00,003C,F9F0,2C00,0000,1221  ;2438  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1221, 081D,2008,0180,0800,0000,1222  ;2439  D_Q-D-1   ; Subtract 1 from total
U 1222, 0000,003C,F980,3C00,0000,10DC  ;2440  ID[3E]_D  ; Adjust the pointer
U 10DC, 0000,003D,0180,0800,0000,11DE  ;2441 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10DD, 0000,003E,0180,0800,0000,0001  ;2442  RETURN1   ; Tell caller another unit was found
            ;2443
```

```
            ;2444  .PAGE
            ;2445  .TOC "   Select Controller"
            ;2446 ;++
            ;2447 ; FUNCTIONAL DESCRIPTION:
            ;2448 ;
            ;2449 ; This routine is used to put the memory system into the
            ;2450 ; specified configuration with respect to controller select.
            ;2451 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2452 ; a RETURN2 is made.
            ;2453 ;
            ;2454 ; CALLING CONSTRAINT:
            ;2455 ;
            ;2456 ; =00
            ;2457 ;
            ;2458 ; INPUTS:
            ;2459 ;
            ;2460 ; d - Contains value to write to bits<2:0> of Register A
            ;2461 ; rc[0e] - Address of Register A
            ;2462 ;
            ;2463 ; SIDE EFFECTS:
            ;2464 ;
            ;2465 ; sc,d,va are destroyed
            ;2466 ;--
            ;2467 SELECT.CNTRLR:
U 1223, 0010,0038,0180,0970,0284,7224  ;2468  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1224, 0801,001C,0180,0800,0000,1225  ;2469  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1225, 0000,003C,0180,A800,0000,1226  ;2470  cache.p_d[long]  ; Write the configuration Register
U 1226, 0818,0038,5D80,0800,0000,1227  ;2471  d_k[.7]   ; Get Misconfiguration bits
U 1227, 0000,003C,61F8,0800,0084,7228  ;2472  sc_k[.F],q_0  ; ...
U 1228, 0D00,003C,0180,0800,0000,1229  ;2473  d_dal.sc  ; ... D = x38000
U 1229, 0000,003C,01E0,0800,0000,122A  ;2474  q_d   ; Save mask
U 122A, 0000,003C,0180,C800,0000,122B  ;2475  d[long]_cache.p  ; Read CNFG A Reg and
            ;2476  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 122B, 001D,2034,0180,0800,0010,122C  ;2477   clk.ubcc ; ...
U 122C, 0000,013C,0180,0800,0000,10DE  ;2478  z?   ; Branch if no
U 10DE, 0000,003E,0180,0800,0000,0001  ;2479 =0 RETURN1   ; Bad configuration
U 10DF, 0000,003E,0180,0800,0000,0002  ;2480  RETURN2   ; Configuration ok
            ;2481
```

```
;2482  .PAGE
;2483  .TOC "    SELECT LOWER CONTROLLER"
;2484 ;*************************************************************
;2485 ;++
;2486 ;
;2487 ; Functional Description:
;2488 ; ----------------------
;2489 ;
;2490 ; This subroutine can be called to select the lower
;2491 ; Controller on a MS780-E/H.
;2492 ; If the Lower controller is misconfigured then a
;2493 ; RETURN1 will be made.  Otherwise a RETURN2
;2494 ;
;2495 ; Calling Constraint: =00
;2496 ; ------------------
;2497 ;
;2498 ;
;2499 ; Inputs:
;2500 ; ------
;2501 ;
;2502 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2503 ;
;2504 ; Outputs:
;2505 ; -------
;2506 ;
;2507 ; RC[0D] will have the address of CNFG Register C
;2508 ;        ( 2000x008 )
;2509 ;
;2510 ; Side Effects:
;2511 ; ------------
;2512 ;
;2513 ; Contents of the following registers will lost:
;2514 ;
;2515 ; D
;2516 ;
;2517 ; The Lower controller on the MS780-E/H will be configured
;2518 ; when the subroutine is exitted with a RETURN2
;2519 ;--
;2520 ;*************************************************************
;2521 =00
;2522 SELECT.LOWER:
;2523   D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1004, 0818,0039,1980,0800,0000,1223  ;2524   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1005, 0000,003E,0180,0800,0000,0001  ;2525  RETURN1     ; Lower Controller Misconfigured
U 1006, 0810,0038,0180,0970,0000,1007  ;2526  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2527  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1007, 0019,0016,0180,09E8,0000,0002  ;2528   RETURN2    ; Let caller know that the controller
;2529       ; ...was configured properly
```

```
          ;2530  .PAGE
          ;2531  .TOC "    SELECT UPPER CONTROLLER"
          ;2532 ;*****************************************************
          ;2533 ;++
          ;2534 ;
          ;2535 ; Functional Description:
          ;2536 ; ----------------------
          ;2537 ;
          ;2538 ; This subroutine can be called to select the upper
          ;2539 ; Controller on a MS780-E/H.
          ;2540 ; If the Upper controller is misconfigured then a
          ;2541 ; RETURN1 will be made.  Otherwise a RETURN2
          ;2542 ;
          ;2543 ; Calling Constraint: =00
          ;2544 ; ------------------
          ;2545 ;
          ;2546 ;
          ;2547 ; Inputs:
          ;2548 ; ------
          ;2549 ;
          ;2550 ; RC[0E] Must hold the I/O base address of the MS780-E/H
          ;2551 ;
          ;2552 ; Outputs:
          ;2553 ; -------
          ;2554 ;
          ;2555 ; RC[0D] will have the address of CNFG Register D
          ;2556 ;        ( 2000x010 )
          ;2557 ;
          ;2558 ; Side Effects:
          ;2559 ; ------------
          ;2560 ;
          ;2561 ; Contents of the following registers will lost:
          ;2562 ;
          ;2563 ; D
          ;2564 ;
          ;2565 ; The Upper controller on the MS780-E/H will be configured
          ;2566 ; when the subroutine is exitted with a RETURN2
          ;2567 ;--
          ;2568 ;*****************************************************
          ;2569 =00
          ;2570 SELECT.UPPER:
          ;2571  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,1223  ;2572   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;2573  RETURN1    ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;2574  D_RC[0E]   ; Get MS780-E/H I/O Base address
          ;2575  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;2576   RETURN2   ; Let caller know that the controller
          ;2577       ; ...was configured properly
```

```
            ;2578  .PAGE
            ;2579  .TOC "   Set Trap Mask"
            ;2580 ;++
            ;2581 ; FUNCTIONAL DESCRIPTION:
            ;2582 ;
            ;2583 ; This routine is used to set a bit in the Micro Trap Mask
            ;2584 ; R[0F].
            ;2585 ;
            ;2586 ; CALLING CONSTRAINT:
            ;2587 ;
            ;2588 ; =0
            ;2589 ;
            ;2590 ; INPUTS:
            ;2591 ;
            ;2592 ; SC - Contains bit number to set.
            ;2593 ;
            ;2594 ; OUTPUTS:
            ;2595 ;
            ;2596 ; rc[0E] - Contains the current trap mask
            ;2597 ;
            ;2598 ; SIDE EFFECTS:
            ;2599 ;
            ;2600 ; d regsiter is destroyed
            ;2601 ;--
            ;2602 SET_TRAP_MASK:
U 122D, 0800,003C,0180,0A78,0000,122E  ;2603  d_r[0f]
U 122E, 0801,001C,0180,0AF8,0000,122F  ;2604  r[0f]_d.ornot.mask,d_alu ; Set mask
U 122F, 0001,003E,0180,0AF0,0000,0001  ;2605  r[0E]_d,return1    ; Save mask and return
            ;2606
```

```
;2607  .PAGE
;2608  .TOC "   CONFIGURE MS780-E/H"
;2609 ;*************************************************************
;2610 ;**
;2611 ;
;2612 ; Functional Description:
;2613 ; ----------------------
;2614 ;
;2615 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2616 ; specifically  check  the  memory, but  make use of it to execute.
;2617 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2618 ; that  if  a  Read/Write operation will take place no false errors
;2619 ; will be logged due to misconfigured memory.
;2620 ;
;2621 ; Calling Constraint: =00
;2622 ; ---------------------
;2623 ;
;2624 ;
;2625 ; Inputs:
;2626 ; ------
;2627 ;
;2628 ; RC[0E] - I/O BASE OF MS780-E/H
;2629 ;
;2630 ;
;2631 ; Outputs:
;2632 ; -------
;2633 ;
;2634 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2635 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2636 ;      OR IF NO MS780-E/Hs WERE FOUND
;2637 ;      ON THE SBI
;2638 ;
;2639 ; Side Effects:
;2640 ; ------------
;2641 ;
;2642 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2643 ;
;2644 ;
;2645 ;
;2646 ;--
;2647 ;*************************************************************
;2648 CONFIG.MS780E:
U 1230, 0818,0038,0D80,0800,0000,1231  ;2649  D_K[.3]     ; Set up flag for the Fail Chain
U 1231, 0001,003C,0180,09E8,0000,1014  ;2650  RC[0D]_D     ; ...
U 1014, 0000,003D,0180,0800,0000,10C0  ;2651 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1015, 0000,003D,0980,0800,0084,70BC  ;2652  ERROR1[.2]    ; No MS780-E/H's found on the SBI
U 1016, 0000,003C,0180,0800,0000,1018  ;2653  NOP     ; OK. Found at least one MS780-E/H
;2654 =
;2655 CONFIG.RETRY:     ; Entry point for the case in which the
;2656       ; ...first MS780-E/H could not be
;2657       ; ...properly configured
U 1018, 0000,003D,0180,0800,0000,1004  ;2658 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1019, 0000,003C,0180,0800,0000,101C  ;2659  J/CNFG.LMIS    ; Lower controller misconfigured
U 101A, 0000,003E,0180,0800,0000,0001  ;2660  RETURN1     ; OK. Lower Controller is configured
;2661 =
```

```
            ;2662 =00
            ;2663 CNFG.LMIS:
U 101C, 0000,003D,0180,0800,0000,1010  ;2664  CALL[SELECT.UPPER]  ; Select the upper controller
U 101D, 0000,003C,0180,0800,0000,1020  ;2665  J/CNFG.UMIS    ; Upper Controller Misconfigured
U 101E, 0000,003E,0180,0800,0000,0001  ;2666  RETURN1     ; OK. Upper Controller is configured
            ;2667 =
            ;2668 =00
            ;2669 CNFG.UMIS:
U 1020, 0000,003D,0180,0800,0000,1215  ;2670  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2671       ; ...MS780-E/H found so go and see
            ;2672       ; ...if there are any more
U 1021, 0000,003C,0180,0800,0000,1232  ;2673  J/CONFIG.RETRY    ; Found another MS780-E/H. Go and
            ;2674       ; ...try to configure it
U 1022, 0818,0038,0D80,0800,0000,1023  ;2675  D_K[.3]     ; Set Flag for Fail Chain
U 1023, 0001,003C,0180,09E8,0000,1232  ;2676  RC[0D]_D    ; ...
U 1232, 0000,003D,0980,0800,0084,70BC  ;2677  ERROR1[.2]    ; Could not configure any MS780-E/H
            ;2678       ; ...abort the test
            ;2679 =
            ;2680
```

```
            ;2681  .PAGE
            ;2682  .TOC "   ENBFLT ROUTINE"
            ;2683 ;*****************************************************************
            ;2684 ;*
            ;2685 ;       THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2686 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2687 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2688 ;
            ;2689 ; D AND SC GET CLOBBERED
            ;2690 ;-
            ;2691 ;*****************************************************************
U 1233, 0F00,003C,2180,0800,0084,7234  ;2692 ENBFLT: SC_K[.14],D_0
U 1234, 0000,003C,0580,0800,0084,B235  ;2693  SC_SC-K[.1]
U 1235, 0801,001C,0180,0800,0000,1236  ;2694  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 1236, 0600,003C,6D80,3C00,0000,1237  ;2695  ID[FAULT]_D,D_D.RIGHT
U 1237, 0000,003E,6D80,3C00,0000,0001  ;2696  ID[FAULT]_D,RETURN1
            ;2697
```

```
                ;2698  .PAGE
                ;2699  .TOC "    NOINTR ROUTINE"
                ;2700 ;*****************************************************************
                ;2701 ;*
                ;2702 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
                ;2703 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
                ;2704 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
                ;2705 ; IF ANY INTERRUPTS OCCURRED:
                ;2706 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
                ;2707 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
                ;2708 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
                ;2709 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
                ;2710 ;  RETURN1 IS USED TO GET BACK
                ;2711 ; ELSE
                ;2712 ;  RETURN2 IS USED
                ;2713 ;
                ;2714 ; THE D AND Q REGISTERS GET CLOBBERED.
                ;2715 ;
                ;2716 ;-
                ;2717 ;*****************************************************************
                ;2718 =0
U 10E0, 0000.003D.0180.0800.0000.123D  ;2719 NOINTR: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 10E1, 0000.003C.35F0.2C00.0000.1238  ;2720  Q_ID[VECTOR]
U 1238, 0019.2034.8180.0800.0010.1239  ;2721  ALU_Q.AND.K[.3FF],CLK.UBCC
U 1239, 0000.013C.0180.0800.0000.10E2  ;2722  Z?
U 10E2, 0000.003C.0180.0800.0000.1241  ;2723 =0 J/ERRFLT    ; INTR. OCCURRED,SET FLAG,GET DATA
U 10E3, 0000.003E.0180.0800.0000.0002  ;2724  RETURN2     ; NO INTR, RETURN2
                ;2725 =0
U 10E4, 0000.003D.0180.0800.0000.123D  ;2726 WRTOUT: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 10E5, 0000.003C.35F0.2C00.0000.123A  ;2727  Q_ID[VECTOR]
U 123A, 0819.2034.8180.0800.0000.123B  ;2728  ALU_Q.AND.K[.3FF],D_ALU
U 123B, 0019.0020.A580.0800.0010.123C  ;2729  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 123C, 0000.013C.0180.0800.0000.10E6  ;2730  Z?
U 10E6, 0000.003C.0180.0800.0000.1241  ;2731 =0 J/ERRFLT    ; WRONG VECTOR,SET FLAG,GET DATA
U 10E7, 0000.003E.0180.0800.0000.0002  ;2732  RETURN2     ; OK, RETURN2
U 123D, 0F00.003C.0180.0800.0000.123E  ;2733 STROBE: D_0      ; SET LOWEST IPL ACTIVE
U 123E, 0000.003C.3D80.3C00.0000.123F  ;2734  ID[PSL]_D    ;
U 123F, 0000.003C.0180.0800.4000.1240  ;2735  IEK/ISTR     ; STROBE INTERRUPT
U 1240, 0000.003E.0180.0800.0000.0001  ;2736  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 1241, 0018.0038.0580.09A8.0000.1242  ;2737 ERRFLT: RC[5]_K[.1]    ; Set Flag
U 1242, 0001.203C.65F0.2DC0.0000.1243  ;2738  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 1243, 0001.203C.6DF0.2DB8.0000.1244  ;2739  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 1244, 0001.203E.0180.09B0.0000.0001  ;2740  RC[6]_Q,RETURN1     ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
                ;2741
                ;2742 ;/////////////////////////////////////////////////////////////////
                ;2743 ;*
                ;2744 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
                ;2745 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
                ;2746 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
                ;2747 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
                ;2748 ;
                ;2749 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
                ;2750 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
                ;2751 ;
                ;2752 ;   =0 D_0,CALL,J/DC5
```

```
            :2753 ;   NOP
            :2754 ;  =0 CALL,J/DC5
            :2755 ;-
            :2756
U 1245, 0018,0038,1980,0800,0000,1255   ;2757 DC0: ALU_0(K),J/S00
U 1246, 0018,0038,1980,0800,0000,1256   ;2758 DC1: ALU_0(K),J/S01
U 1247, 0018,0038,1980,0800,0000,1257   ;2759 DC2: ALU_0(K),J/S10
U 1248, 0018,0038,1980,0800,0000,1258   ;2760 DC3: ALU_0(K),J/S11
U 1249, 0018,0038,0980,0800,0000,1255   ;2761 DC4: ALU_K[.2],J/S00
U 124A, 0018,0038,0980,0800,0000,1256   ;2762 DC5: ALU_K[.2],J/S01
U 124B, 0018,0038,0980,0800,0000,1257   ;2763 DC6: ALU_K[.2],J/S10
U 124C, 0018,0038,0980,0800,0000,1258   ;2764 DC7: ALU_K[.2],J/S11
U 124D, 0018,0038,0580,0800,0000,1255   ;2765 DC8: ALU_K[.1],J/S00
U 124E, 0018,0038,0580,0800,0000,1256   ;2766 DC9: ALU_K[.1],J/S01
U 124F, 0018,0038,0580,0800,0000,1257   ;2767 DCA: ALU_K[.1],J/S10
U 1250, 0018,0038,0580,0800,0000,1258   ;2768 DCB: ALU_K[.1],J/S11
U 1251, 0018,0038,C180,0800,0000,1255   ;2769 DCC: ALU_K[.FFFF],J/S00
U 1252, 0018,0038,C180,0800,0000,1256   ;2770 DCD: ALU_K[.FFFF],J/S01
U 1253, 0018,0038,C180,0800,0000,1257   ;2771 DCE: ALU_K[.FFFF],J/S10
U 1254, 0018,0038,C180,0800,0000,1258   ;2772 DCF: ALU_K[.FFFF],J/S11
            :2773
U 1255, 0118,0038,1B80,0800,0000,1259   ;2774 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1256, 0118,0038,0B80,0800,0000,1259   ;2775 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 1257, 0118,0038,0780,0800,0000,1259   ;2776 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 1258, 0118,0038,C380,0800,0000,1259   ;2777 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
U 1259, 0100,003E,0380,0800,0000,0001   ;2778 SX: D_D.LEFT2,S'/7,RETURN1 ; EXIT
            :2779
            :2780
            :2781
U 125A, 0818,0038,0180,0800,0000,105E   ;2782 DELAY: D_K[.8],J/CALLCON
            :2783
```

```
            ;2784  .PAGE
            ;2785  .TOC "   Wait Loop Routine"
            ;2786 ;++
            ;2787 ; FUNCTIONAL DESCRIPTION:
            ;2788 ;
            ;2789 ; This routine is used to wait the specified number of machine cycles.
            ;2790 ; This routine is typically called to wait for an SBI write to
            ;2791 ; I/O space to complete.
            ;2792 ;
            ;2793 ; CALLING CONSTRAINT:
            ;2794 ;
            ;2795 ; =0
            ;2796 ;
            ;2797 ; INPUTS:
            ;2798 ;
            ;2799 ; d - Contains number of cycles to wait
            ;2800 ; alu.z condition code must not be set
            ;2801 ;
            ;2802 ; SIDE EFFECTS:
            ;2803 ;
            ;2804 ; d is destroyed
            ;2805 ;--
            ;2806 WAIT.LOOP:
U 125B, 0018,0038,6180,0800,0010,10E8  ;2807  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2808      ; ...is not set
            ;2809 =0
            ;2810 W.LOOP:
            ;2811  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10E8, 0819,0100,0580,0800,0010,10E8  ;2812       j/W.LOOP
U 10E9, 0000,003E,0180,0800,0000,0001  ;2813  return1   ; exit
            ;2814
```

```
            ;2815  .PAGE
            ;2816  .TOC "    Initialize the SBI"
            ;2817 ;++
            ;2818 ; FUNCTIONAL DESCRIPTION:
            ;2819 ;
            ;2820 ; This routine performs the following functions:
            ;2821 ;
            ;2822 ;  Executes an SBI Unjam
            ;2823 ;  Clears the SBI Fault Latch
            ;2824 ;  Clears the SBI Error Register
            ;2825 ;
            ;2826 ; CALLING CONSTRAINT:
            ;2827 ;
            ;2828 ; =0
            ;2829 ;
            ;2830 ; SIDE EFFECTS:
            ;2831 ;
            ;2832 ; D & Q Register destroyed
            ;2833 ;--
            ;2834 =0
            ;2835 SBI.INIT:
U 10EA, 0000.003D.0180.0800.0000.1260  ;2836  call[sbi.unjam]  ; Unjam the SBI
U 10EB, 0818.0038.C180.0800.0000.125C  ;2837  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 125C, 0801.0028.6580.3C00.0000.125D  ;2838  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 125D, 0F00.003C.6D80.3C00.0000.125E  ;2839  id[fault]_d,d_0
U 125E, 0000.003C.6D80.3C00.0000.125F  ;2840  id[fault]_d
U 125F, 0000.003E.6580.3C00.0000.0001  ;2841  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2842
            ;2843
            ;2844 SBI.UNJAM:
            ;2845 CLRSBI:
            ;2846 ;============================================================================
            ;2847 ;////////////////////////////////////////////////////////////////////////////
            ;2848 ;+
            ;2849 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2850 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2851 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2852 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2853 ;-
U 1260, 0818.0038.6580.8000.0010.10EC  ;2854 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2855 =0
U 10EC, 0819.0100.0580.8000.0010.10EC  ;2856 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 10ED, 0818.0038.6580.8800.0010.10EE  ;2857  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2858 =0
U 10EE, 0819.0100.0580.8800.0010.10EE  ;2859 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 10EF, 0818.0038.6580.8000.0010.10F0  ;2860  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2861 =0
U 10F0, 0819.0100.0580.8000.0010.10F0  ;2862 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 10F1, 0000.003E.0180.0800.0000.0001  ;2863  RETURN1    ; DONE.
            ;2864
```

```
            ;2865   .PAGE
            ;2866   .TOC "    INIT ROUTINE"
            ;2867 INIT: ;=================================================================
            ;2868   ;
            ;2869   ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2870   ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2871   ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2872   ;
            ;2873   ;=================================================================
            ;2874   ;
U 1261, 0003.003C.0180.0AF8.0000.10F2   ;2875   R[0F]_0     ;CLEAR U-TRAPS
U 10F2, 0000.003D.0180.0800.0000.1260   ;2876  =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 10F3, 0000.003C.0180.0800.0000.10F4   ;2877   NOP
U 10F4, 0000.003D.0180.0800.0000.1264   ;2878  =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2879         ;  INTRUPT REG
U 10F5, 0000.003C.0180.0800.0000.10F6   ;2880   NOP
U 10F6, 0000.003D.0180.0800.0000.126B   ;2881  =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 10F7, 0000.003C.0180.0800.0000.10F8   ;2882   NOP
U 10F8, 0000.003D.0180.0800.0000.126F   ;2883  =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 10F9, 0000.003C.0180.0800.0000.10FA   ;2884   NOP
U 10FA, 0000.003D.0180.0800.0000.1267   ;2885  =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 10FB, 0000.003C.0180.0800.0000.10FC   ;2886   NOP
U 10FC, 0000.003D.0180.0800.0000.1273   ;2887  =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 10FD, 0000.003C.0180.0800.0000.10FE   ;2888   NOP
U 10FE, 0000.003D.0180.0800.0000.1233   ;2889  =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 10FF, 0818.0038.4580.0800.0000.110A   ;2890   D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;2891  =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 110A, 0000.003D.7980.3C00.0000.11DF   ;2892    ID[PARITY]_D
U 110B, 0F00.003C.0180.0800.0000.1262   ;2893   D_0
U 1262, 0000.003C.4980.3C00.0000.1263   ;2894   ID[TBER0]_D    ;SHUT TB OFF
U 1263, 0000.003E.7180.3C00.0000.0001   ;2895   ID[COMP]_D,RETURN1   ;DISABLE SILO INTERRUPTS
            ;2896   ;
            ;2897 SETPSL: ;=================================================================
            ;2898   ;
            ;2899   ;DROP THE CPU IPR LEVEL TO
            ;2900   ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2901   ; PENDING
            ;2902   ;
            ;2903   ;=================================================================
            ;2904   ;
U 1264, 0F00.003C.0180.0800.0000.1265   ;2905   D_0
U 1265, 0000.003C.3980.3C00.0000.1266   ;2906   ID[SIR]_D   ;SET SOFTWARE LEVELS TO ZERO
U 1266, 0000.003E.3D80.3C00.0000.0001   ;2907   RETURN1,ID[PSL]_D
            ;2908 CLRTIM: ;=================================================================
            ;2909   ;
            ;2910   ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2911   ; SBI ERROR REGISTER
            ;2912   ;
            ;2913   ;=================================================================
            ;2914   ;
U 1267, 0818.0038.0580.0800.0000.1268   ;2915   D_K[.1]
U 1268, 0000.003C.85F8.0800.0084.7269   ;2916   Q_0,SC_K[.C]
U 1269, 0D00.003C.0180.0800.0000.126A   ;2917   D_DAL.SC
U 126A, 0000.003E.6580.3C00.0000.0001   ;2918   ID[SBI.ERR]_D,RETURN1
            ;2919 CLRFLT: ;=================================================================
```

```
            ;2920   ;
            ;2921   ;CLEAR THE CP FAULT BIT IN THE
            ;2922   ; FAULT REGISTER
            ;2923   ;
            ;2924   ;=============================================================================
            ;2925   ;
U 126B, 0818.0038.0180.0800.0000.126C   ;2926   D_K[.8]
U 126C, 0000.003C.65F8.0800.0084.726D   ;2927   Q_0,SC_K[.10]
U 126D, 0D00.003C.0180.0800.0000.126E   ;2928   D_DAL.SC
U 126E, 0000.003E.6D80.3C00.0000.0001   ;2929   ID[FAULT]_D,RETURN1
            ;2930 CLRECC: ;=============================================================================
            ;2931   ;
            ;2932   ;CLEAR CRD,RDS BIT IN THE
            ;2933   ;SBI ERROR REGISTER
            ;2934   ;
            ;2935   ;=============================================================================
            ;2936   ;
U 126F, 0818.0038.0D80.0800.0000.1270   ;2937   D_K[.3]
U 1270, 0000.003C.89F8.0800.0084.7271   ;2938   Q_0,SC_K[.D]
U 1271, 0D00.003C.0180.0800.0000.1272   ;2939   D_DAL.SC
U 1272, 0000.003E.6580.3C00.0000.0001   ;2940   ID[SBI.ERR]_D,RETURN1
            ;2941 ENBECC: ;=============================================================================
            ;2942   ;
            ;2943   ;ENABLE CRD,RDS INTERRUPTS
            ;2944   ;
            ;2945   ;=============================================================================
            ;2946   ;
U 1273, 0818.0038.0580.0800.0000.1274   ;2947   D_K[.1]
U 1274, 0000.003C.61F8.0800.0084.7275   ;2948   Q_0,SC_K[.F]
U 1275, 0D00.003C.0180.0800.0000.1276   ;2949   D_DAL.SC
U 1276, 0000.003E.6580.3C00.0000.0001   ;2950   ID[SBI.ERR]_D,RETURN1
            ;2951
```

```
;2952   .PAGE "TEST 1D9 CANCEL TEST FOR SBI CYCLES "
;2953   ;
;2954   ;*****************************************************************************
;2955   ;**
;2956   ;
;2957   ; CANCEL TEST FOR SBI OPERATIONS
;2958   ;
;2959   ; TEST DESCRIPTION
;2960   ;
;2961   ; THIS TEST CHECK TO SEE THAT SBI OPERATIONS ARE CONCELLED
;2962   ; FOR CERTAIN CPU ERRORS. THE ERROR WILL BE FORCED DURING
;2963   ; A WRITE OPERATION. THE DATA WILL BE CHECKED IN MEMORY TO
;2964   ; SEE THAT IT IS UNCHANGED. THE U-TRAP WILL BE CHECKED FOR
;2965   ; ALSO.
;2966   ;
;2967   ; CURRENT FUNTIONS TESTED FOR CANCELING OF SBI CYCLES
;2968   ; ====================================================
;2969   ;  TEST SECTION  DESCRIPTION
;2970   ;  ************  ***********
;2971   ;    SUBTEST 01  INITIALIZE ONE TIME OPERATIONS
;2972   ;    SUBTEST 02  NORMAL WRITES, NO ERRORS FORCED
;2973   ;      MCT/WRITE.V.WCHK
;2974   ;    TB-SBI CYCLES CHECKED ( WRITE.V.WCHK )
;2975   ;    SUBTEST 03  NORMAL READ,NO ERRORS FORCED
;2976   ;    TB-SBI CYCLES CHECKED (READ.V.RCHK )
;2977   ;    SUBTEST 04  CANCEL: TB MISS ON WRITES
;2978   ;     MCT/WRITE.V.WCHK
;2979   ;    SUBTEST 05  CANCEL: TB MISS ON READS, U-TRAPS ONLY
;2980   ;     MCT/READ.V.RCHK
;2981   ;     JUST CHECKS TO SEE U-TRAP
;2982   ;    SUBTEST 06  TB PARITY ERROR: GROUP 0, DATA 0
;2983   ;    SUBTEST 07  TB PARITY ERROR: GROUP 0, ADDRESS 0
;2984   ;    SUBTEST 08  TB PARITY ERROR: GROUP 1, DATA 0
;2985   ;    SUBTEST 09  TB PARITY ERROR: GROUP 1, ADDRESS 0
;2986   ;    SUBTEST A  M BIT U-TRAP
;2987   ;     WRITE.V.WCHK WITH M BIT = 0
;2988   ;    SUBTEST B  PROTECTION U-TRAP
;2989   ;        SET PROTECTION TO NO ACCESS AND
;2990   ;     USE MCT/WRITE.V.WCHK
;2991   ;    SUBTEST C  PAGE BOUNDRY U-TRAP
;2992   ;     SET TO EDGE OF PAGE AND INITIATE
;2993   ;     LONG WORD WRITE OPERATION
;2994   ;    SUBTEST D  CANCELLING OF STALL
;2995   ;     SET A CONDITION THAT TIMEOUTS
;2996   ;     AND INITIATE A TB MISS, SEE NO TIMEOUT
;2997   ;
;2998   ; LOGIC DESCRIPTION
;2999   ;
;3000   ; THE SBI INTERFACE BOARDS SBIL(M8218), SBLH(M8219)
;3001   ; IF NO UTRAP THEN THE FAULTS LIE WITHIN THE TB,TRAPS
;3002   ; AND INTERRUPTS,OR U-SEQUENCER BOARDS.
;3003   ;
;3004   ; ERROR LOGOUT
;3005   ;
;3006   ; DATA: EXPECTED DATA
```

```
;3007 ;  DATA READ AFTER THE ATTEMPTED VIRTUAL OPERATION
;3008 ;  UNDEFINED
;3009 ;  UNDEFINED
;3010 ;  UNDEFINED
;3011 ;  UNDEFINED
;3012 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3013 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3014 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3015 ;  INTERRUPT FLAG:
;3016 ;   0=>NO INTERRUPT OCCURRED
;3017 ;   1=>AN INTERRUPT OCCURRED
;3018 ;
;3019 ;
;3020 ; NOTE: SEE SUBTEST DESCRIPTION FOR MORE DETAIL
;3021 ;
;3022 ;SYNC POINT DESCRIPTION
;3023 ;
;3024 ;--
;3025 ;
;3026 ;
;3027 ;================================================================================
;3028 ;
;3029 ; SCRATCH PAD ASSIGNMENTS
;3030 ;
;3031 ;
;3032 ; RA DESCRIPTION
;3033 ; 0 TEST DATA -1,^XFFFFFFFF
;3034 ; 1 FORCE TB REPLACE GROUP 0   ^X40000
;3035 ; 2 FORCE TB REPLACE GROUP 1   ^X80000
;3036 ; 3 BASE: FORCE PARITY ERROR GROUP 0, DATA 0 ^X5
;3037 ; 4 BASE: FORCE PARITY ERROR GROUP 0, ADDR 0 ^X11
;3038 ; 5 BASE: FORCE PARITY ERROR GROUP 1, DATA 0 ^XB
;3039 ; 6 BASE: FORCE PARITY ERROR GROUP 1, ADDR 0 ^X17
;3040 ; 7 PTE VALUE FOR TAG 0,ALLOWS ALL ACCESS MODES
;3041 ;   TRANSLATES VA TO PA PFN 0 ^XA400 0000
;3042 ; 8 CONSTANT TO SHUT OFF ECC IN MEMORY ^X100
;3043 ; 9 CONSTANT TO FORCE MISSES IN CACHE ^X18000
;3044 ; E TEMPERARY STARAGE
;3045 ; F U-TRAP MASK
;3046 ;
;3047 ;
;3048 ; RC DESCRIPTION
;3049 ; ====================================
;3050 ; 0 CONFIGURATION REG A ADDRESS   ^X2000 2000
;3051 ; 1 CONFIGURATION REG B ADDRESS   ^X2000 2004
;3052 ; 2 CONFIGURATION REG C ADDRESS   ^X2000 2008
;3053 ; 3 TEST ADDRESS LOCATION (PHYSICAL ADDRESS )  ^X100
;3054 ; 4 VIRTUAL ADDRESS DISPLACEMENT (BASE PFN)    ^X200
;3055 ; 5 VIRTUAL TEST ADRESS RC 3 + RC 4
;3056 ; F # OF ARG TO CONSOLE    ^X0
;3057 ;
;3058 ;
;3059 ;********************************************************************************
;3060 ;
;3061 ;
```

```
            ;3062 ; TO CHECK
            ;3063 ; ========================
            ;3064 ; CACHE PARITY ERRORS
            ;3065 ; IB CHECKS
            ;3066 ;
            ;3067 ;--
            ;3068 ;*****************************************************************************
            ;3069 ;
            ;3070 =0
U 1110, 0000.003D.0180.0800.0000.104F  ;3071 C06000: NEWTST
U 1111, 0018.0038.F580.09F8.0000.1112  ;3072  RC[0F]_K[.A]
U 1112, 0000.003D.0180.0800.0000.1230  ;3073 =0 CALL[CONFIG.MS780E]   ;
U 1113, 0000.003C.0180.0800.0000.127D  ;3074  J/C06005
            ;3075 TBVAL: ;=================================================================
            ;3076  ;*
            ;3077  ; VALIDATE TB ENTRY 2
            ;3078  ;USES RC 4 AS THE VIRTUAL PFN
            ;3079  ;      RA 7 AS THE PTE
            ;3080  ;VIRTUAL PFN 2 = PHYSICAL PFN 0
            ;3081  ;FORCES REPLACE BOTH IN TBER0 ID REGISTER
            ;3082  ;-
            ;3083  ;=================================================================
            ;3084  ;
U 1277, 0F00.003C.2180.0800.0084.7278  ;3085  D_0.SC_K[.14]
U 1278, 0801.001C.0180.0800.0000.1279  ;3086  D_D.ORNOT.MASK   ;SET FORCE RELPACE BOTH
U 1279, 0010.0038.4980.3D20.0200.127A  ;3087  ID[TBER0]_D.VA_RC[04]
U 127A, 0800.003C.0180.0A38.0000.127B  ;3088  D_R[07]    ;GET PTE
U 127B, 0000.003C.4180.3C00.0000.127C  ;3089  ID[TBUF]_D   ;WRITE THE ENTRY
U 127C, 0000.003E.0180.0800.0000.0001  ;3090  RETURN1
            ;3091 ;*****************************************************************************
            ;3092 ;*
            ;3093 ; SUBTEST 00
            ;3094 ; INITIALIZE THE ONE
            ;3095 ; TIME CONSTANTS
            ;3096 ;-
            ;3097 ;*****************************************************************************
            ;3098 ;
U 127D, 0F00.003C.6580.0800.0084.727E  ;3099 C06005: D_0,SC_K[.10]   ;RA 1
U 127E, 0000.003C.0980.0800.0084.927F  ;3100  SC_SC*K[.2]
U 127F, 0801.001C.0180.0800.0000.1280  ;3101  D_D.ORNOT.MASK
U 1280, 0001.003C.0180.0A88.0000.1281  ;3102  R[1]_D
U 1281, 0500.003C.0180.0800.0000.1282  ;3103  D_D.LEFT   ;RA 2
U 1282, 0001.003C.0180.0A90.0000.1283  ;3104  R[2]_D
U 1283, 0F00.003C.0180.0800.0000.1114  ;3105  D_0    ;RA 3
U 1114, 0000.003D.0180.0800.0000.124A  ;3106 =0 CALL[DCS]
U 1115, 0001.003C.0180.0A98.0000.1284  ;3107  R[3]_D
U 1284, 0818.0038.0580.0800.0000.1285  ;3108  D_K[.1]    ;RA 4
U 1285, 0000.003C.11F8.0800.0084.7286  ;3109  SC_K[.4],Q_0
U 1286, 0D00.003C.0180.0800.0000.1287  ;3110  D_DAL.SC
U 1287, 0819.0030.0580.0AA0.0000.1288  ;3111  R[4]_ALU,D_D.OR.K[.1]
U 1288, 0F00.003C.0180.0800.0000.1116  ;3112  D_0    ;RA 5
U 1116, 0000.003D.0180.0800.0000.1250  ;3113 =0 CALL[DCB]
U 1117, 0001.003C.0180.0AA8.0000.1289  ;3114  R[5]_D
U 1289, 0818.0038.0580.0800.0000.128A  ;3115  D_K[.1]    ;RA 6
U 128A, 0000.003C.11F8.0800.0084.728B  ;3116  SC_K[.4],Q_0
```

```
U 128B, 0D00.003C.0180.0800.0000.128C  ;3117  D_DAL.SC
U 128C, 0819.0030.5D80.0AB0.0000.128D  ;3118  R[6]_ALU,D_D.OR.K[.7]
U 128D, 0F00.003C.0180.0800.0000.1118  ;3119  D_0     ;RA 7
U 1118, 0000.003D.0180.0800.0000.124F  ;3120 =0 CALL[DCA]
U 1119, 0000.003C.11F8.0800.0084.728E  ;3121  SC_K[.4],Q_0
U 128E, 0D00.003C.0180.0800.0000.128F  ;3122  D_DAL.SC
U 128F, 0819.0030.1180.0800.0000.1290  ;3123  D_D.OR.K[.4]
U 1290, 0000.003C.7DF8.0800.0084.7291  ;3124  Q_0,SC_K[.18]   ;24 SHIFTS
U 1291, 0D00.003C.0180.0800.0000.1292  ;3125  D_DAL.SC
U 1292, 0001.003C.0180.0AB8.0000.1293  ;3126  R[7]_D
U 1293, 0F00.003C.0180.0800.0084.7294  ;3127  D_0,SC_K[.8]   ;RA 8
U 1294, 0801.001C.0180.0800.0000.1295  ;3128  D_D.ORNOT.MASK
U 1295, 0001.003C.0180.0AC0.0000.1296  ;3129  R[8]_D
U 1296, 0818.0038.8580.0800.0000.1297  ;3130  D_K[.C]     ;RA 9
U 1297, 0000.003C.89F8.0800.0084.7298  ;3131  SC_K[.D],Q_0
U 1298, 0D00.003C.0180.0800.0000.1299  ;3132  D_DAL.SC
U 1299, 0001.003C.0180.0AC8.0000.129A  ;3133  R[9]_D
U 129A, 0F00.003C.0180.0800.0084.729B  ;3134  D_0,SC_K[.8]   ;RC 3
U 129B, 0801.001C.0180.0998.0000.129C  ;3135  RC[3]_ALU,D_D.ORNOT.MASK
U 129C, 0500.003C.0180.0800.0000.129D  ;3136  D_D.LEFT    ;RC 4
U 129D, 0001.003C.0180.09A0.0000.129E  ;3137  RC[4]_D
U 129E, 0F00.003C.0180.0800.0000.129F  ;3138  D_0     ;RA 0
U 129F, 0801.0028.0180.0800.0000.111A  ;3139  D_NOT.D
U 111A, 0001.003C.0180.0A80.0000.111B  ;3140 =0 R[0]_D
U 111B, 0010.0038.01C0.091B.0000.12A0  ;3141  Q_RC[03]    ;GET VIRTUAL TEST ADDRESS
U 12A0, 0810.0038.0180.0920.0000.12A1  ;3142  D_RC[04]
U 12A1, 001D.0014.0180.09A8.0000.111C  ;3143  RC[05]_ALU,ALU_D[A+B]Q
          ;3144 ;
          ;3145 ;*********************************************************************************
          ;3146 ;*
          ;3147 ;SUBTEST 01
          ;3148 ;WRITE OPERATION USING THE TB,NO ERRORS FORCED
          ;3149 ;  VIRTUAL PFN 1 = PHYSICAL PFN 0
          ;3150 ;-
          ;3151 ;*********************************************************************************
          ;3152 ;
U 111C, 0000.003D.0180.0800.0000.1154  ;3153 =0 CALL[SUBTEST1]
U 111D, 0000.003C.0180.0800.0000.111E  ;3154  NOP
U 111E, 0000.003D.0180.0800.0000.10A7  ;3155 =0 ERLOOP
U 111F, 0000.003C.0180.0800.0000.1120  ;3156  NOP
U 1120, 0000.003D.0180.0800.0000.1261  ;3157 =0 CALL[INIT]
U 1121, 0000.003C.0180.0800.0000.1122  ;3158  NOP
U 1122, 0000.003D.0180.0800.0000.1277  ;3159 =0 CALL[TBVAL]
U 1123, 0F10.0038.0180.0918.0200.12A2  ;3160  D_0,VA_RC[03]   ;WRITE BACKGROUND PATTERN
U 12A2, 0000.003C.0180.A800.0000.1124  ;3161  MCT/WRITE.P
U 1124, 0818.0039.D580.0800.0000.125B  ;3162 =0 D_K[.6],CALL[WAIT.LOOP]
U 1125, 0000.003C.0180.0800.0000.1024  ;3163  NOP
U 1024, 0000.003D.0180.0800.0000.10E0  ;3164 =00 CALL[NOINTR]
U 1025, 0000.003D.0180.0800.0000.10AB  ;3165  ERROR1
U 1026, 0818.0038.0580.0800.0000.12A3  ;3166  D_K[.1]     ;ENABLE MEMORY MANAGEMENT
U 12A3, 0000.003C.4980.3C00.0000.12A4  ;3167 = ID[TBER0]_D
U 12A4, 0800.003C.0180.0A00.0000.12A5  ;3168  D_R[0]
U 12A5, 0001.003C.0180.09F0.0000.12A6  ;3169  RC[0E]_D    ;SET EXPECTED DATA
U 12A6, 0010.0038.0180.0928.0200.12A7  ;3170  VA_RC[05]
U 12A7, 0000.003C.0180.3000.0000.1126  ;3171  MCT/WRITE.V.WCHK
```

```
U 1126. 0818.0039.D580.0800.0000.125B  ;3172 =0 D_K[.6],CALL[WAIT.LOOP]
U 1127. 0000.003C.0180.0800.0000.1028  ;3173  NOP
U 1028. 0000.003D.0180.0800.0000.10E0  ;3174 =00 CALL[NOINTR]
U 1029. 0000.003D.0180.0800.0000.10AB  ;3175  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 102A. 0010.0038.0180.0918.0200.12A8  ;3176  VA_RC[03]   ;SEE IF DATA WAS CHANGEDD
U 12A8. 0000.003C.0180.C800.0000.12A9  ;3177 = MCT/READ.P
U 12A9. 0001.003C.0180.09E8.0000.102C  ;3178  RC[0D]_D
U 102C. 0000.003D.0180.0800.0000.10E0  ;3179 =00 CALL[NOINTR]
U 102D. 0000.003D.0180.0800.0000.10AB  ;3180  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 102E. 0810.0038.0180.0968.0000.12AB  ;3181  D_RC[0D]
U 12AB. 000D.0020.0180.0A00.0010.12AC  ;3182 = ALU_D[XOR]R[0],CLK.UBCC  ;TEST
U 12AC. 0000.013C.0180.0800.0000.1128  ;3183  Z?
U 1128. 0000.003D.0180.0800.0000.10AB  ;3184 =0 ERROR1     ;DATA WAS DIFFERENT
U 1129. 0000.003C.0180.0800.0000.112A  ;3185  NOP
            ;3186 ;
            ;3187 ;*****************************************************************************
            ;3188 ;*
            ;3189 ;SUBTEST 02
            ;3190 ;READ OPERATION USING THE TB,NO ERRORS FORCED
            ;3191 ; VIRTUAL PFN 1 = PHYSICAL PFN 0
            ;3192 ;-
            ;3193 ;*****************************************************************************
            ;3194 ;
U 112A. 0000.003D.0180.0800.0000.10BB  ;3195 =0 SUBTEST
U 112B. 0000.003C.0180.0800.0000.112C  ;3196  NOP
U 112C. 0000.003D.0180.0800.0000.10A7  ;3197 =0 ERLOOP
U 112D. 0000.003C.0180.0800.0000.112E  ;3198  NOP
U 112E. 0000.003D.0180.0800.0000.1261  ;3199 =0 CALL[INIT]
U 112F. 0000.003C.0180.0800.0000.1130  ;3200  NOP
U 1130. 0000.003D.0180.0800.0000.1277  ;3201 =0 CALL[TBVAL]
U 1131. 0000.003C.0180.0800.0000.12AD  ;3202  NOP
U 12AD. 0F10.0038.0180.0918.0200.12AE  ;3203  D_0,VA_RC[03]   ;WRITE BACKGROUND DATA
U 12AE. 0000.003C.0180.A800.0C00.1132  ;3204  MCT/WRITE.P
U 1132. 0818.0039.D580.0800.0000.125B  ;3205 =0 D_K[.6],CALL[WAIT.LOOP]
U 1133. 0000.003C.0180.0800.0000.1030  ;3206  NOP
U 1030. 0000.003D.0180.0800.0000.10E0  ;3207 =00 CALL[NOINTR]
U 1031. 0000.003D.0180.0800.0000.10AB  ;3208  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1032. 0818.0038.0580.0800.0000.12AF  ;3209  D_K[.1]     ;SET MME BIT
U 12AF. 0000.003C.4980.3C00.0000.12B0  ;3210 = ID[TBER0]_D
U 12B0. 0010.0038.0180.0928.0200.12B1  ;3211  VA_RC[05]
U 12B1. 0800.003C.0180.0A00.0000.12B2  ;3212  D_R[0]
U 12B2. 0001.003C.0180.09F0.0000.12B3  ;3213  RC[0E]_D    ;SET EXPECTED DATA
U 12B3. 0000.003C.0180.3000.0000.1134  ;3214  MCT/WRITE.V.WCHK
U 1134. 0818.0039.D580.0800.0000.125B  ;3215 =0 D_K[.6],CALL[WAIT.LOOP]
U 1135. 0000.003C.0180.0800.0000.1034  ;3216  NOP
U 1034. 0000.003D.0180.0800.0000.10E0  ;3217 =00 CALL[NOINTR]
U 1035. 0000.003D.0180.0800.0000.10AB  ;3218  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 1036. 0F10.0038.0180.0928.0200.12B4  ;3219  VA_RC[05],D_0    ;CHECK DATA WRITTEN USING A
U 12B4. 0000.003C.0180.4000.0000.12B5  ;3220 = MCT/READ.V.RCHK    ;    VIRTUAL OPERATION
U 12B5. 0001.003C.0180.09E8.0000.1038  ;3221  RC[0D]_D
U 1038. 0000.003D.0180.0800.0000.10E0  ;3222 =00 CALL[NOINTR]
U 1039. 0000.003D.0180.0800.0000.10AB  ;3223  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 103A. 0810.0038.0180.0968.0000.12B6  ;3224  D_RC[0D]
U 12B6. 000D.0020.0180.0A00.0010.12B7  ;3225 = ALU_D[XOR]R[0],CLK.UBCC  ;TEST
U 12B7. 0000.013C.0180.0800.0000.1136  ;3226  Z?
```

```
U 1136, 0000,003D,0180,0800,0000,10AB  ;3227 =0 ERROR1     ;DATA WAS DIFFERENT
U 1137, 0000,003C,0180,0800,0000,1138  ;3228  NOP
          ;3229 ;
          ;3230 ;*********************************************************************************
          ;3231 ;*
          ;3232 ;SUBTEST 03
          ;3233 ;CANCEL ON A TB MISS(WRITE )
          ;3234 ;USE INVALID ENTRY AS MISS
          ;3235 ;-
          ;3236 ;*********************************************************************************
          ;3237
U 1138, 0000,003D,0180,0800,0000,10BB  ;3238 =0 SUBTEST
U 1139, 0000,003C,0180,0800,0000,113A  ;3239  NOP
U 113A, 0000,003D,0180,0800,0000,10A7  ;3240 =0 ERLOOP
U 113B, 0000,003C,0180,0800,0000,113C  ;3241  NOP
U 113C, 0000,003D,0180,0800,0000,1261  ;3242 =0 CALL[INIT]
U 113D, 0000,003C,0180,0800,0000,113E  ;3243  NOP
U 113E, 0000,003D,0180,0800,0000,1277  ;3244 =0 CALL[TBVAL]   ;SET TB VALID
          ;3245       ;INVALIDATE TB TAG 0
U 113F, 0F00,003C,2180,0800,0084,72B8  ;3246  D_0,SC_K[.14]   ;SET MASK FOR REPLACE BOTH
U 12B8, 0801,001C,0180,0800,0000,12B9  ;3247  D_D.ORNOT.MASK   ;GENERATE  FORCE REPLACE "BOTH"
U 12B9, 0000,003C,4980,3C00,0000,12BA  ;3248  ID[TBER0]_D   ;WRITE THE FORCE REPLACE CONSTANT
U 12BA, 0F10,0038,0180,0920,0200,12BB  ;3249  VA_RC[04],D_0   ;SET THE PTE
U 12BB, 0000,003C,4180,3C00,0000,12BC  ;3250  ID[TBUF]_D   ;WRITE THE PTE
U 12BC, 0F10,0038,0180,0918,0200,12BD  ;3251  D_0,VA_RC[03]   ;WRITE THE BACKGROUND DATA
U 12BD, 0003,003C,0180,09F0,0000,12BE  ;3252  RC[0E]_0   ;SET EXPECTED DATA
U 12BE, 0000,003C,0180,A800,0000,1140  ;3253  MCT/WRITE.P
U 1140, 0818,0039,D580,0800,0000,125B  ;3254 =0 D_K[.6],CALL[WAIT.LOOP]
U 1141, 0000,003C,0180,0800,0000,103C  ;3255  NOP
U 103C, 0000,003D,0180,0800,0000,10E0  ;3256 =00 CALL[NOINTR]
U 103D, 0000,003D,0180,0800,0000,10AB  ;3257  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 103E, 0818,0038,0580,0800,0000,12BF  ;3258  D_K[.1]    ;SET MME BIT
U 12BF, 0000,003C,4980,3C00,0000,12C0  ;3259 = ID[TBER0]_D
U 12C0, 0010,0038,0180,0928,0200,12C1  ;3260  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 12C1, 0F00,003C,D580,0800,0084,72C2  ;3261  SC_K[.6],D_0   ;SET U-TRAP FLAG
U 12C2, 0801,001C,0180,0AF8,0000,12C3  ;3262  R[0F]_ALU,D_D.ORNOT.MASK
U 12C3, 0800,003C,0180,0A00,0000,12C4  ;3263  D_R[0]
U 12C4, 0000,003C,0180,3000,0000,1142  ;3264  MCT/WRITE.V.WCHK
U 1142, 0818,0039,D580,0800,0000,125B  ;3265 =0 D_K[.6],CALL[WAIT.LOOP]  ;SHOULD U-TRAP HERE
U 1143, 0000,003C,0180,0800,0000,1040  ;3266  NOP
U 1040, 0000,003D,0180,0800,0000,10E0  ;3267 =00 CALL[NOINTR]
U 1041, 0000,003D,0180,0800,0000,10AB  ;3268  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1042, 0000,003C,0180,0A78,0010,12C5  ;3269  ALU_R[0F],CLK.UBCC  ;CHECK U-TRAP FLAG
U 12C5, 0000,013C,0180,0800,0000,1144  ;3270 = Z?
U 1144, 0000,003D,0180,0800,0000,10AB  ;3271 =0 ERROR1     ;U-TRAP DID NOT OCCURE
          ;3272       ;CHECK DATA IN MEMORY FOR NO CHANGE
U 1145, 0010,0038,0180,0918,0200,12C6  ;3273  VA_RC[03]   ;LOAD THE ADDRESS
U 12C6, 0000,003C,0180,C800,0000,12C7  ;3274  MCT/READ.P
U 12C7, 0001,003C,0180,09E8,0000,1044  ;3275  RC[0D]_D
U 1044, 0000,003D,0180,0800,0000,10E0  ;3276 =00 CALL[NOINTR]
U 1045, 0000,003D,0180,0800,0000,10AB  ;3277  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1046, 0810,0038,0180,0968,0000,12C8  ;3278  D_RC[0D]
U 12C8, 0001,003C,0180,0800,0010,12C9  ;3279 = ALU_D,CLK.UBCC    ;TEST
U 12C9, 0000,013C,0180,0800,0000,1146  ;3280  Z?
U 1146, 0000,003D,0180,0800,0000,10AB  ;3281 =0 ERROR1     ;DATA WAS DIFFERENT
```

```
U 1147, 0000.003C.0180.0800.0000.1148  ;3282  NOP
        ;3283 ;
        ;3284 ;*********************************************************************************
        ;3285 ;*
        ;3286 ;SUBTEST 04
        ;3287 ;CANCEL ON A TB MISS(READ )
        ;3288 ;U-TRAP CHECK ONLY
        ;3289 ;-
        ;3290 ;*********************************************************************************
        ;3291 ;
U 1148, 0000.003D.0180.0800.0000.10BB  ;3292 =0 SUBTEST
U 1149, 0000.003C.0180.0800.0000.114A  ;3293  NOP
U 114A, 0000.003D.0180.0800.0000.10A7  ;3294 =0 ERLOOP
U 114B, 0000.003C.0180.0800.0000.114C  ;3295  NOP
U 114C, 0000.003D.0180.0800.0000.1261  ;3296 =0 CALL[INIT]
U 114D, 0000.003C.0180.0800.0000.114E  ;3297  NOP
U 114E, 0000.003D.0180.0800.0000.1277  ;3298 =0 CALL[TBVAL]   ;VALIDATE THE TB
        ;3299       ;INVALIDATE TB TAG 0
U 114F, 0F00.003C.2180.0800.0084.72CA  ;3300  D_0,SC_K[.14]   ;SET THE MASK FOR THE FORCE
        ;3301       ; REPLACE BOTH BIT
U 12CA, 0801.001C.0180.0800.0000.12CB  ;3302  D_D.ORNOT.MASK   ;GENERATE THE FORCE REPLACE
        ;3303       ;  ENTRY
U 12CB, 0000.003C.4980.3C00.0000.12CC  ;3304  ID[TBER0]_D   ;SET THE ID REGISTER WITH
        ;3305       ;  FORCE REPLACE BITS
U 12CC, 0F10.0038.0180.0920.0200.12CD  ;3306  VA_RC[04],D_0   ;SET THE PTE
U 12CD, 0000.003C.4180.3C00.0000.12CE  ;3307  ID[TBUF]_D   ;WRITE THE PTE
U 12CE, 0F10.0038.0180.0918.0200.12CF  ;3308  D_0,VA_RC[03]   ;SET BACKGROUND DATA
U 12CF, 0000.003C.0180.A800.0000.1150  ;3309  MCT/WRITE.P
U 1150, 0818.0039.D580.0800.0000.125B  ;3310 =0 D_K[.6],CALL[WAIT.LOOP]
U 1151, 0000.003C.0180.0800.0000.1048  ;3311  NOP
U 1048, 0000.003D.0180.0800.0000.10E0  ;3312 =00 CALL[NOINTR]
U 1049, 0000.003D.0180.0800.0000.10AB  ;3313  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 104A, 0818.0038.0580.0800.0000.12D0  ;3314  D_K[.1]    ;SET MME BIT
U 12D0, 0000.003C.4980.3C00.0000.12D1  ;3315 = ID[TBER0]_D
U 12D1, 0010.0038.0180.0928.0200.12D2  ;3316  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL READ
U 12D2, 0F00.003C.D580.0800.0084.72D3  ;3317  SC_K[.6],D_0   ;SET U-TRAP FLAG
U 12D3, 0801.001C.0180.0AF8.0000.12D4  ;3318  R[0F]_ALU,D_D.ORNOT.MASK
U 12D4, 0000.003C.0180.4000.0000.12D5  ;3319  MCT/READ.V.RCHK   ;READ CAUSING U-TRAP
U 12D5, 0001.003C.0180.09E8.0000.1050  ;3320  RC[0D]_D
U 1050, 0000.003D.0180.0800.0000.10E0  ;3321 =00 CALL[NOINTR]
U 1051, 0000.003D.0180.0800.0000.10AB  ;3322  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1052, 0810.0038.0180.0968.0000.12D6  ;3323  D_RC[0D]
U 12D6, 0000.003C.0180.0A78.0010.12D7  ;3324 = ALU_R[0F],CLK.UBCC  ;CHECK FOR U-TRAP
U 12D7, 0000.013C.0180.0800.0000.1152  ;3325  Z?
U 1152, 0000.003D.0180.0800.0000.10AB  ;3326 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 1153, 0000.003C.0180.0800.0000.1156  ;3327  NOP
        ;3328 ;
        ;3329 ;*********************************************************************************
        ;3330 ;*
        ;3331 ;SUBTEST 05
        ;3332 ;CANCEL FOR TB PARITY ERROR,GROUP 0,DATA 0
        ;3333 ;-
        ;3334 ;*********************************************************************************
        ;3335 ;
U 1156, 0000.003D.0180.0800.0000.10BB  ;3336 =0 SUBTEST
```

```
U 1157. 0000.003C.0180.0800.0000.1158  ;3337  NOP
U 1158. 0000.003D.0180.0800.0000.10A7  ;3338 =0 ERLOOP
U 1159. 0000.003C.0180.0800.0000.115A  ;3339  NOP
U 115A. 0000.003D.0180.0800.0000.1261  ;3340 =0 CALL[INIT]
U 115B. 0000.003C.0180.0800.0000.115C  ;3341  NOP
U 115C. 0000.003D.0180.0800.0000.1277  ;3342 =0 CALL[TBVAL]
U 115D. 0800.003C.0180.0A08.0000.12D8  ;3343  D_R[1]     ;FORCE REPLACE
U 12D8. 0000.003C.4980.3C00.0000.12D9  ;3344  ID[TBER0]_D
U 12D9. 0010.0038.0180.0920.0200.12DA  ;3345  VA_RC[04]   ;WRITE PTE ENTRY
U 12DA. 0800.003C.0180.0A38.0000.12DB  ;3346  D_R[7]
U 12DB. 0000.003C.4180.3C00.0000.12DC  ;3347  ID[TBUF]_D
U 12DC. 0F10.0038.0180.0918.0200.12DD  ;3348  D_0,VA_RC[03]   ;WRITE BACKGROUND DATA
U 12DD. 0003.003C.0180.09F0.0000.12DE  ;3349  RC[0E]_0   ;SET EXPECTED DATA
U 12DE. 0000.003C.0180.AB00.0000.115E  ;3350  MCT/WRITE.P
U 115E. 0818.0039.D580.0800.0000.125B  ;3351 =0 D_K[.6],CALL[WAIT.LOOP]
U 115F. 0000.003C.0180.0800.0000.1054  ;3352  NOP
U 1054. 0000.003D.0180.0800.0000.10E0  ;3353 =00 CALL[NOINTR]
U 1055. 0000.003D.0180.0800.0000.10AB  ;3354  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1056. 0800.003C.0180.0A18.0000.12DF  ;3355  D_R[03]    ;SET FORCE PARITY ERROR
U 12DF. 0819.0030.0580.0800.0000.12E0  ;3356 = D_D.OR.K[.1]   ;SET MME BIT
U 12E0. 0000.003C.4980.3C00.0000.12E1  ;3357  ID[TBER0]_D
U 12E1. 0010.0038.0180.0928.0200.12E2  ;3358  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 12E2. 0F00.003C.5D80.0800.0084.72E3  ;3359  SC_K[.7],D_0   ;SET U-TRAP MASK
U 12E3. 0801.001C.0180.0AF8.0000.12E4  ;3360  R[0F]_ALU,D_D.ORNOT.MASK
U 12E4. 0800.003C.0180.0A00.0000.12E5  ;3361  D_R[0]
U 12E5. 0000.003C.0180.3000.0000.1160  ;3362  MCT/WRITE.V.WCHK  ;VIRTUAL OPERATION
U 1160. 0818.0039.D580.0800.0000.125B  ;3363 =0 D_K[.6],CALL[WAIT.LOOP]
U 1161. 0000.003C.0180.0800.0000.1060  ;3364  NOP
U 1060. 0000.003D.0180.0800.0000.10E0  ;3365 =00 CALL[NOINTR]
U 1061. 0000.003D.0180.0800.0000.10AB  ;3366  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1062. 0000.003C.0180.0A78.0010.12E6  ;3367  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 12E6. 0000.013C.0180.0800.0000.1162  ;3368 = Z?
U 1162. 0000.003D.0180.0800.0000.10AB  ;3369 =0 ERROR1    ;U-TRAP DID NOT OCCURE
          ;3370       ;CHECK THE MEMORY FOR NO CHANGE
          ;3371       ;    IN THE CONTENTS
U 1163. 0010.0038.0180.0918.0200.12E7  ;3372  VA_RC[03]   ;LOAD THE ADDRESS
U 12E7. 0000.003C.0180.C800.0000.12E8  ;3373  MCT/READ.P
U 12E8. 0001.003C.0180.09E8.0000.1064  ;3374  RC[0D]_D   ;SAVE DATA READ
U 1064. 0000.003D.0180.0800.0000.10E0  ;3375 =00 CALL[NOINTR]
U 1065. 0000.003D.0180.0800.0000.10AB  ;3376  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1066. 0010.0038.0180.0968.0010.12E9  ;3377  ALU_RC[0D],CLK.UBCC  ;TEST
U 12E9. 0000.013C.0180.0800.0000.1164  ;3378 = Z?
U 1164. 0000.003D.0180.0800.0000.10AB  ;3379 =0 ERROR1    ;DATA WAS DIFFERENT
U 1165. 0000.003C.0180.0800.0000.1166  ;3380  NOP
          ;3381 ;
          ;3382 ;*****************************************************************************
          ;3383 ;*
          ;3384 ;SUBTEST 06
          ;3385 ;CANCEL FOR TB PARITY ERROR GROUP 0, ADDRESS 0
          ;3386 ;-
          ;3387 ;*****************************************************************************
          ;3388 ;
U 1166. 0000.003D.0180.0800.0000.10BB  ;3389 =0 SUBTEST
U 1167. 0000.003C.0180.0800.0000.1168  ;3390  NOP
U 1168. 0000.003D.0180.0800.0000.10A7  ;3391 =0 ERLOOP
```

```
U 1169, 0000.003C.0180.0800.0000.116A  ;3392  NOP
U 116A, 0000.003D.0180.0800.0000.1261  ;3393 =0 CALL[INIT]
U 116B, 0000.003C.0180.0800.0000.116C  ;3394  NOP
U 116C, 0000.003D.0180.0800.0000.1277  ;3395 =0 CALL[TBVAL]
U 116D, 0000.003C.0180.0800.0000.12EA  ;3396  NOP
U 12EA, 0800.003C.0180.0A08.0000.12EB  ;3397  D_R[1]    ;FORCE REPLACE
U 12EB, 0000.003C.4980.3C00.0000.12EC  ;3398  ID[TBER0]_D
U 12EC, 0010.0038.0180.0920.0200.12ED  ;3399  VA_RC[04]   ;WRITE PTE ENTRY
U 12ED, 0800.003C.0180.0A38.0000.12EE  ;3400  D_R[7]
U 12EE, 0000.003C.4180.3C00.0000.12EF  ;3401  ID[TBUF]_D
U 12EF, 0F10.0038.0180.0918.0200.12F0  ;3402  D_0,VA_RC[03]   ;WRITE THE BACKGROUND DATA
U 12F0, 0003.003C.0180.09F0.0000.12F1  ;3403  RC[0E]_0   ;SET EXPECTED DATA
U 12F1, 0000.003C.0180.A800.0000.116E  ;3404  MCT/WRITE.P
U 116E, 0818.0039.D580.0800.0000.125B  ;3405 =0 D_K[.6],CALL[WAIT.LOOP]
U 116F, 0000.003C.0180.0800.0000.1068  ;3406  NOP
U 1068, 0000.003D.0180.0800.0000.10E0  ;3407 =00 CALL[NOINTR]
U 1069, 0000.003D.0180.0800.0000.10AB  ;3408  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 106A, 0800.003C.0180.0A20.0000.12F2  ;3409  D_R[4]    ;GET FORCE PARITY CONSTANT
U 12F2, 0819.0030.0580.0800.0000.12F3  ;3410 = D_ALU,ALU_D[OR]K[.1]  ;SET MME BIT
U 12F3, 0000.003C.4980.3C00.0000.12F4  ;3411  ID[TBER0]_D   ;WRITE THE ID REGISTER
U 12F4, 0010.0038.0180.0928.0200.12F5  ;3412  VA_RC[05]   ;LOAD ADDRESS FOR VIRTUAL REFERENCE
U 12F5, 0F00.003C.5D80.0800.0084.72F6  ;3413  SC_K[.7],D_0   ;SET U-TRAP MASK
U 12F6, 0801.001C.0180.0AF8.0000.12F7  ;3414  R[0F]_ALU,D_D.ORNOT.MASK
U 12F7, 0800.003C.0180.0A00.0000.12F8  ;3415  D_R[0]
U 12F8, 0000.003C.0180.3000.0000.1170  ;3416  MCT/WRITE.V.WCHK  ;VIRTUAL OPERATION
U 1170, 0818.0039.D580.0800.0000.125B  ;3417 =0 D_K[.6],CALL[WAIT.LOOP]
U 1171, 0000.003C.0180.0800.0000.106C  ;3418  NOP
U 106C, 0000.003D.0180.0800.0000.10E0  ;3419 =00 CALL[NOINTR]
U 106D, 0000.003D.0180.0800.0000.10AB  ;3420  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 106E, 0000.003C.0180.0A78.0010.12F9  ;3421  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 12F9, 0000.013C.0180.0800.0000.1172  ;3422 = Z?
U 1172, 0000.003D.0180.0800.0000.10AB  ;3423 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 1173, 0010.0038.0180.0918.0200.12FA  ;3424  VA_RC[03]   ;CHECK FOR NO CHANGE IN MEMORY
U 12FA, 0000.003C.0180.C800.0000.12FB  ;3425  MCT/READ.P
U 12FB, 0001.003C.0180.09E8.0000.1070  ;3426  RC[0D]_D   ;SAVE DATA READ
U 1070, 0000.003D.0180.0800.0000.10E0  ;3427 =00 CALL[NOINTR]
U 1071, 0000.003D.0180.0800.0000.10AB  ;3428  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1072, 0010.0038.0180.0968.0010.12FC  ;3429  ALU_RC[0D],CLK.UBCC  ;TEST
U 12FC, 0000.013C.0180.0800.0000.1174  ;3430 = Z?
U 1174, 0000.003D.0180.0800.0000.10AB  ;3431 =0 ERROR1    ;DATA WAS DIFFERENT
U 1175, 0000.003C.0180.0800.0000.1176  ;3432  NOP
          ;3433 ;
          ;3434 ;*****************************************************************************
          ;3435 ;*
          ;3436 ;SUBTEST 07
          ;3437 ;CANCEL FOR TB PARITY ERROR,GROUP 1,DATA 0
          ;3438 ;-
          ;3439 ;*****************************************************************************
          ;3440 ;
U 1176, 0000.003D.0180.0800.0000.10BB  ;3441 =0 SUBTEST
U 1177, 0000.003C.0180.0800.0000.1178  ;3442  NOP
U 1178, 0000.003D.0180.0800.0000.10A7  ;3443 =0 ERLOOP
U 1179, 0000.003C.0180.0800.0000.117A  ;3444  NOP
U 117A, 0000.003D.0180.0800.0000.1261  ;3445 =0 CALL[INIT]
U 117B, 0000.003C.0180.0800.0000.117C  ;3446  NOP
```

```
U 117C. 0000,003D,0180,0800,0000,1277  ;3447 =0 CALL[TBVAL]
U 117D. 0000,003C,0180,0800,0000,12FD  ;3448  NOP
U 12FD. 0800,003C,0180,0A10,0000,12FE  ;3449  D_R[2]    ;FORCE REPLACE GROUP 1
U 12FE. 0000,003C,4980,3C00,0000,12FF  ;3450  ID[TBER0]_D
U 12FF. 0010,0038,0180,0920,0200,1300  ;3451  VA_RC[04]   ;WRITE PTE ENTRY
U 1300. 0800,003C,0180,0A38,0000,1301  ;3452  D_R[7]
U 1301. 0000,003C,4180,3C00,0000,1302  ;3453  ID[TBUF]_D
U 1302. 0F10,0038,0180,0918,0200,1303  ;3454  D_0,VA_RC[03]   ;WRITE BACKGROUND DATA
U 1303. 0003,003C,0180,09F0,0000,1304  ;3455  RC[0E]_0   ;SET EXPECTED DATA
U 1304. 0000,003C,0180,A800,0000,117E  ;3456  MCT/WRITE.P
U 117E. 0818,0039,D580,0800,0000,125B  ;3457 =0 D_K[.6],CALL[WAIT.LOOP]
U 117F. 0000,003C,0180,0800,0000,1074  ;3458  NOP
U 1074. 0000,003D,0180,0800,0000,10E0  ;3459 =00 CALL[NOINTR]
U 1075. 0000,003D,0180,0800,0000,10AB  ;3460  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1076. 0800,003C,0180,0A28,0000,1077  ;3461  D_R[5]    ;FORCE PARITY ERROR
U 1077. 0819,0030,0580,0800,0000,1305  ;3462  D_D.OR.K[.1]   ;ENABLE MEMORY MAMAGEMENT
U 1305. 0000,003C,4980,3C00,0000,1306  ;3463  ID[TBER0]_D
U 1306. 0010,0038,0180,0928,0200,1307  ;3464  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 1307. 0F00,003C,5D80,0800,0084,7308  ;3465  SC_K[.7],D_0   ;SET U-TRAP MASK
U 1308. 0801,001C,0180,0AF8,0000,1309  ;3466  R[0F]_ALU,D_D.ORNOT.MASK
U 1309. 0800,003C,0180,0A00,0000,130A  ;3467  D_R[0]
U 130A. 0000,003C,0180,3000,0000,1180  ;3468  MCT/WRITE.V.WCHK  ;VIRTUAL REFERENCE
U 1180. 0818,0039,D580,0800,0000,125B  ;3469 =0 D_K[.6],CALL[WAIT.LOOP]
U 1181. 0000,003C,0180,0800,0000,1078  ;3470  NOP
U 1078. 0000,003D,0180,0800,0000,10E0  ;3471 =00 CALL[NOINTR]
U 1079. 0000,003D,0180,0800,0000,10AB  ;3472  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 107A. 0000,003C,0180,0A78,0010,130B  ;3473  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 130B. 0000,013C,0180,0800,0000,1182  ;3474 = Z?
U 1182. 0000,003D,0180,0800,0000,10AB  ;3475 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 1183. 0010,0038,0180,0918,0200,130C  ;3476  VA_RC[03]   ;CHECK DATA IN MEMORY
U 130C. 0000,003C,0180,C800,0000,130D  ;3477  MCT/READ.P
U 130D. 0001,003C,0180,09E8,0000,107C  ;3478  RC[0D]_D   ;SAVE DATA READ
U 107C. 0000,003D,0180,0800,0000,10E0  ;3479 =00 CALL[NOINTR]
U 107D. 0000,003D,0180,0800,0000,10AB  ;3480  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 107E. 0010,0038,0180,0968,0010,130E  ;3481  ALU_RC[0D],CLK.UBCC  ;TEST
U 130E. 0000,013C,0180,0800,0000,1184  ;3482 = Z?
U 1184. 0000,003D,0180,0800,0000,10AB  ;3483 =0 ERROR1    ;DATA WAS DIFFERENT
U 1185. 0000,003C,0180,0800,0000,1186  ;3484  NOP
          ;3485 ;
          ;3486 ;*****************************************************************************
          ;3487 ;*
          ;3488 ;SUBTEST 08
          ;3489 ;CANCEL FOR TB PARITY ERROR GROUP 1, ADDRESS 0
          ;3490 ;-
          ;3491 ;*****************************************************************************
          ;3492 ;
U 1186. 0000,003D,0180,0800,0000,10BB  ;3493 =0 SUBTEST
U 1187. 0000,003C,0180,0800,0000,1188  ;3494  NOP
U 1188. 0000,003D,0180,0800,0000,10A7  ;3495 =0 ERLOOP
U 1189. 0000,003C,0180,0800,0000,118A  ;3496  NOP
U 118A. 0000,003D,0180,0800,0000,1261  ;3497 =0 CALL[INIT]
U 118B. 0000,003C,0180,0800,0000,118C  ;3498  NOP
U 118C. 0000,003D,0180,0800,0000,1277  ;3499 =0 CALL[TBVAL]
U 118D. 0800,003C,0180,0A10,0000,130F  ;3500  D_R[2]    ;FORCE REPLACE
U 130F. 0000,003C,4980,3C00,0000,1310  ;3501  ID[TBER0]_D
```

```
U 1310. 0010,0038,0180,0920,0200,1311  ;3502  VA_RC[04]   ;WRITE PTE ENTRY
U 1311. 0800,003C,0180,0A38,0000,1312  ;3503  D_R[7]
U 1312. 0000,003C,4180,3C00,0000,1313  ;3504  ID[TBUF]_D
U 1313. 0F10,0038,0180,0918,0200,1314  ;3505  D_0,VA_RC[03]   ;WRITE BACKGROUND DATA
U 1314. 0003,003C,0180,09F0,0000,1315  ;3506  RC[0E]_0   ;SET EXPECTED DATA
U 1315. 0000,003C,0180,A800,0000,118E  ;3507  MCT/WRITE.P
U 118E. 0818,0039,D580,0800,0000,125B  ;3508 =0 D_K[.6],CALL[WAIT.LOOP]
U 118F. 0000,003C,0180,0800,0000,1080  ;3509  NOP
U 1080. 0000,003D,0180,0800,0000,10E0  ;3510 =00 CALL[NOINTR]
U 1081. 0000,003D,0180,0800,0000,10AB  ;3511  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1082. 0800,003C,0180,0A30,0000,1316  ;3512  D_R[6]    ;FORCE PARITY CONSTANT
U 1316. 0819,0030,0580,0800,0000,1317  ;3513 = D_ALU,ALU_D[OR]K[.1]  ;SET MME
U 1317. 0000,003C,4980,3C00,0000,1318  ;3514  ID[TBER0]_D   ;SET ID REGISTER
U 1318. 0010,0038,0180,0928,0200,1319  ;3515  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 1319. 0F00,003C,5D80,0800,0084,731A  ;3516  SC_K[.7],D_0   ;SET U-TRAP MASK
U 131A. 0801,001C,0180,0AF8,0000,131B  ;3517  R[0F]_ALU,D_D.ORNOT.MASK
U 131B. 0800,003C,0180,0A00,0000,131C  ;3518  D_R[0]
U 131C. 0000,003C,0180,3000,0000,1190  ;3519  MCT/WRITE.V.WCHK  ;VIRTUAL REFERENCE
U 1190. 0818,0039,D580,0800,0000,125B  ;3520 =0 D_K[.6],CALL[WAIT.LOOP]
U 1191. 0000,003C,0180,0800,0000,1084  ;3521  NOP
U 1084. 0000,003D,0180,0800,0000,10E0  ;3522 =00 CALL[NOINTR]
U 1085. 0000,003D,0180,0800,0000,10AB  ;3523  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1086. 0000,003C,0180,0A78,0010,131D  ;3524  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 131D. 0000,013C,0180,0800,0000,1192  ;3525 = Z?
U 1192. 0000,003D,0180,0800,0000,10AB  ;3526 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 1193. 0010,0038,0180,0918,0200,131E  ;3527  VA_RC[03]   ;CHECK DATA IN MEMORY
U 131E. 0000,003C,0180,C800,0000,131F  ;3528  MCT/READ.P
U 131F. 0001,003C,0180,09E8,0000,1088  ;3529  RC[0D]_D   ;SAVE DATA READ
U 1088. 0000,003D,0180,0800,0000,10E0  ;3530 =00 CALL[NOINTR]
U 1089. 0000,003D,0180,0800,0000,10AB  ;3531  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 108A. 0010,0038,0180,0968,0010,1320  ;3532  ALU_RC[0D],CLK.UBCC  ;TEST
U 1320. 0000,013C,0180,0800,0000,1194  ;3533 = Z?
U 1194. 0000,003D,0180,0800,0000,10AB  ;3534 =0 ERROR1    ;DATA WAS DIFFERENT
U 1195. 0000,003C,0180,0800,0000,1196  ;3535  NOP
          ;3536 ;
          ;3537 ;*****************************************************************************
          ;3538 ;*
          ;3539 ;SUBTEST 09
          ;3540 ;M BIT U-TRAP CANCEL
          ;3541 ;-
          ;3542 ;*****************************************************************************
          ;3543 ;
U 1196. 0000,003D,0180,0800,0000,10BB  ;3544 =0 SUBTEST
U 1197. 0000,003C,0180,0800,0000,1198  ;3545  NOP
U 1198. 0000,003D,0180,0800,0000,10A7  ;3546 =0 ERLOOP
U 1199. 0000,003C,0180,0800,0000,119A  ;3547  NOP
U 119A. 0000,003D,0180,0800,0000,1261  ;3548 =0 CALL[INIT]
U 119B. 0000,003C,0180,0800,0000,119C  ;3549  NOP
U 119C. 0000,003D,0180,0800,0000,1277  ;3550 =0 CALL[TBVAL]
U 119D. 0010,0038,0180,0920,0200,1321  ;3551  VA_RC[04]   ;WRITE PTE ENTRY
          ;3552          ;  WITH THE M BIT CLEAR
U 1321. 0000,003C,E580,0800,0084,7322  ;3553  SC_K[.1A]   ;CLEAR BIT #26
U 1322. 0000,003C,01C0,0A38,0000,1323  ;3554  Q_R[7]
U 1323. 0801,2034,0180,0800,0000,1324  ;3555  D_Q.AND.MASK
U 1324. 0000,003C,4180,3C00,0000,1325  ;3556  ID[TBUF]_D   ;ENTER PTE
```

```
U 1325. 0F10.0038.0180.0918.0200.1326  ;3557  D_0,VA_RC[03]   ;CLEAR BACKGROUND LOCATION
U 1326. 0003.003C.0180.09F0.0000.1327  ;3558  RC[0E]_0   ;SET EXPECTED DATA
U 1327. 0000.003C.0180.A800.0000.119E  ;3559  MCT/WRITE.P
U 119E. 0818.0039.D580.0800.0000.125B  ;3560 =0 D_K[.6],CALL[WAIT.LOOP]
U 119F. 0000.003C.0180.0800.0000.108C  ;3561  NOP
U 108C. 0000.003D.0180.0800.0000.10E0  ;3562 =00 CALL[NOINTR]
U 108D. 0000.003D.0180.0800.0000.10AB  ;3563  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 108E. 0818.0038.0580.0800.0000.1328  ;3564  D_K[.1]    ;SET MME BIT
U 1328. 0000.003C.4980.3C00.0000.1329  ;3565 = ID[TBER0]_D
U 1329. 0010.0038.0180.0928.0200.132A  ;3566  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 132A. 0F00.003C.0D80.0800.0084.732B  ;3567  SC_K[.3],D_0   ;SET U-TRAP FLAG
U 132B. 0801.001C.0180.0AF8.0000.132C  ;3568  R[0F]_ALU,D_D.ORNOT.MASK
U 132C. 0800.003C.0180.0A00.0000.132D  ;3569  D_R[0]
U 132D. 0000.003C.0180.3000.0000.11A0  ;3570  MCT/WRITE.V.WCHK  ;VIRTUAL REFERENCE
U 11A0. 0818.0039.D580.0800.0000.125B  ;3571 =0 D_K[.6],CALL[WAIT.LOOP]
U 11A1. 0000.003C.0180.0800.0000.1090  ;3572  NOP
U 1090. 0000.003D.0180.0800.0000.10E0  ;3573 =00 CALL[NOINTR]
U 1091. 0000.003D.0180.0800.0000.10AB  ;3574  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1092. 0000.003C.0180.0A78.0010.132E  ;3575  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 132E. 0000.013C.0180.0800.0000.11A2  ;3576 = Z?
U 11A2. 0000.003D.0180.0800.0000.10AB  ;3577 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 11A3. 0010.0038.0180.0918.0200.132F  ;3578  VA_RC[03]   ;CHECK MEMORY CONTENTS
U 132F. 0000.003C.0180.C800.0000.1330  ;3579  MCT/READ.P
U 1330. 0001.003C.0180.09E8.0000.1094  ;3580  RC[0D]_D   ;SAVE DATA READ
U 1094. 0000.003D.0180.0800.0000.10E0  ;3581 =00 CALL[NOINTR]
U 1095. 0000.003D.0180.0800.0000.10AB  ;3582  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 1096. 0010.0038.0180.0968.0010.1331  ;3583  ALU_RC[0D],CLK.UBCC  ;TEST
U 1331. 0000.013C.0180.0800.0000.11A4  ;3584 = Z?
U 11A4. 0000.003D.0180.0800.0000.10AB  ;3585 =0 ERROR1    ;DATA WAS DIFFERENT
U 11A5. 0000.003C.0180.0800.0000.11A6  ;3586  NOP
        ;3587 ;
        ;3588 ;*****************************************************************************
        ;3589 ;+
        ;3590 ;SUBTEST A
        ;3591 ;PROTECTION VIOLATION CANCEL
        ;3592 ;-
        ;3593 ;*****************************************************************************
        ;3594 ;
U 11A6. 0000.003D.0180.0800.0000.10BB  ;3595 =0 SUBTEST
U 11A7. 0000.003C.0180.0800.0000.11A8  ;3596  NOP
U 11A8. 0000.003D.0180.0800.0000.10A7  ;3597 =0 ERLOOP
U 11A9. 0000.003C.0180.0800.0000.11AA  ;3598  NOP
U 11AA. 0000.003D.0180.0800.0000.1261  ;3599 =0 CALL[INIT]
U 11AB. 0000.003C.0180.0800.0000.11AC  ;3600  NOP
U 11AC. 0000.003D.0180.0800.0000.1277  ;3601 =0 CALL[TBVAL]
        ;3602       ;MASK OFF PROTECTION CODE
U 11AD. 0000.003C.ED80.0800.0084.7332  ;3603  SC_K[.1B]   ;SET SHIFT COUNT
U 1332. 0818.0038.61F8.0800.0000.1333  ;3604  D_K[.F],Q_0   ;LOAD 4 BIT MASK
U 1333. 0D00.003C.0180.0800.0000.1334  ;3605  D_DAL.SC   ;SHIFT TO PROTECTION CODE POSITION
U 1334. 081C.2024.0180.0A38.0000.1335  ;3606  D_ALU,ALU_LA[ANDNOT]D,LAB_R[7] ;MASK
U 1335. 0010.0038.0180.0920.0200.1336  ;3607  VA_RC[04]   ;SET ADDTESS
U 1336. 0000.003C.4180.3C00.0000.1337  ;3608  ID[TBUF]_D   ;WRITE THE PTE
U 1337. 0F10.0038.0180.0918.0200.1338  ;3609  D_0,VA_RC[03]   ;WRITE BACKGROUND DATA
U 1338. 0003.003C.0180.09F0.0000.1339  ;3610  RC[0E]_0   ;SET EXPECTED DATA
U 1339. 0000.003C.0180.A800.0000.11AE  ;3611  MCT/WRITE.P
```

```
U 11AE. 0818.0039.D580.0800.0000.125B  ;3612 =0 D_K[.6],CALL[WAIT.LOOP]
U 11AF. 0000.003C.0180.0800.0000.1098  ;3613  NOP
U 1098. 0000.003D.0180.0800.0000.10E0  ;3614 =00 CALL[NOINTR]
U 1099. 0000.003D.0180.0800.0000.10AB  ;3615  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 109A. 0818.0038.0580.0800.0000.133A  ;3616  D_K[.1]    ;SET MME
U 133A. 0000.003C.4980.3C00.0000.133B  ;3617 = ID[TBER0]_D
U 133B. 0010.0038.0180.0928.0200.133C  ;3618  VA_RC[05]   ;SET ADDRESS FOR VIRTUAL REFERENCE
U 133C. 0F00.003C.1180.0800.0084.733D  ;3619  SC_K[.4],D_0   ;SET U-TRAP FLAG
U 133D. 0801.001C.0180.0AF8.0000.133E  ;3620  R[0F]_ALU,D_D.ORNOT.MASK
U 133E. 0800.003C.0180.0A00.0000.133F  ;3621  D_R[0]
U 133F. 0000.003C.0180.3000.0000.11B0  ;3622  MCT/WRITE.V.WCHK
U 11B0. 0818.0039.D580.0800.0000.125B  ;3623 =0 D_K[.6],CALL[WAIT.LOOP]
U 11B1. 0000.003C.0180.0800.0000.109C  ;3624  NOP
U 109C. 0000.003D.0180.0800.0000.10E0  ;3625 =00 CALL[NOINTR]
U 109D. 0000.003D.0180.0800.0000.10AB  ;3626  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 109E. 0000.003C.0180.0A78.0010.1340  ;3627  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 1340. 0000.013C.0180.0800.0000.11B2  ;3628 = Z?
U 11B2. 0000.003D.0180.0800.0000.10AB  ;3629 =0 ERROR1    ;U-TRAP DID NOT OCCURE
U 11B3. 0010.0038.0180.0918.0200.1341  ;3630  VA_RC[03]   ;WRITE BACKGROUND DATA
U 1341. 0000.003C.0180.C800.0000.1342  ;3631  MCT/READ.P
U 1342. 0001.003C.0180.09E8.0000.10A0  ;3632  RC[0D]_D   ;SAVE DATA READ
U 10A0. 0000.003D.0180.0800.0000.10E0  ;3633 =00 CALL[NOINTR]
U 10A1. 0000.003D.0180.0800.0000.10AB  ;3634  ERROR1    ;UNEXPECTED INTERRUPT !!!!!!!!
U 10A2. 0010.0038.0180.0968.0010.1343  ;3635  ALU_RC[0D],CLK.UBCC  ;TEST
U 1343. 0000.013C.0180.0800.0000.11B4  ;3636 = Z?
U 11B4. 0000.003D.0180.0800.0000.10AB  ;3637 =0 ERROR1    ;DATA WAS DIFFERENT
U 11B5. 0000.003C.0180.0800.0000.11B6  ;3638  NOP
          ;3639 ;
          ;3640 ;*****************************************************************************
          ;3641 ;*
          ;3642 ;SUBTEST B
          ;3643 ;CANCEL ON A PAGE BOUNDRY U-TRAP
          ;3644 ;-
          ;3645 ;*****************************************************************************
          ;3646 ;
U 11B6. 0000.003D.0180.0800.0000.10BB  ;3647 =0 SUBTEST
U 11B7. 0000.003C.0180.0800.0000.11B8  ;3648  NOP
U 11B8. 0000.003D.0180.0800.0000.10A7  ;3649 =0 ERLOOP
U 11B9. 0000.003C.0180.0800.0000.11BA  ;3650  NOP
U 11BA. 0000.003D.0180.0800.0000.1261  ;3651 =0 CALL[INIT]
U 11BB. 0000.003C.0180.0800.0000.11BC  ;3652  NOP
U 11BC. 0000.003D.0180.0800.0000.1277  ;3653 =0 CALL[TBVAL]
U 11BD. 0000.003C.0180.0800.0000.1344  ;3654  NOP
U 1344. 0800.003C.0180.0A10.0000.1345  ;3655  D_R[2]    ;FORCE REPLACE GROUP 1
U 1345. 0000.003C.4980.3C00.0000.1346  ;3656  ID[TBER0]_D
U 1346. 0010.0038.0180.0920.0200.1347  ;3657  VA_RC[04]   ;SET ADDRESS FOR PTE ENTRY
U 1347. 0800.003C.0180.0A38.0000.1348  ;3658  D_R[7]
U 1348. 0000.003C.4180.3C00.0000.1349  ;3659  ID[TBUF]_D   ;WRITE PTE ENTRY
U 1349. 0010.0038.49C0.0918.0000.134A  ;3660  Q_RC[03],K[.FF]   ;WRITE BACKGROUND DATA
U 134A. 0F19.2014.49C0.0800.0000.134B  ;3661  D_0,Q_Q+K[.FF]   ;ADDRESS TO PAGE EDGE
U 134B. 0019.2000.0DC0.09E0.0200.134C  ;3662  RC[0C]_ALU,VA_ALU,Q_Q-K[.3] ;LAST LONG WORD BOUNDRY OF PAGE
U 134C. 0000.003C.0180.A800.0000.11BE  ;3663  MCT/WRITE.P   ;WRITE LOWER BACKGROUND DATA
U 11BE. 0818.0039.D580.0800.0000.125B  ;3664 =0 D_K[.6],CALL[WAIT.LOOP]
U 11BF. 0000.003C.0180.0800.0000.10A4  ;3665  NOP
U 10A4. 0000.003D.0180.0800.0000.10E0  ;3666 =00 CALL[NOINTR]
```

```
U 10A5. 0000.003D.0180.0800.0000.10AB  ;3667  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 10A6. 0F00.003C.0180.0803.0000.134D  ;3668  D_0,VA_VA+4   ;WRITE UPPER BACKGROUND DATA
U 134D. 0000.003C.0180.A800.0000.11C0  ;3669 = MCT/WRITE.P
U 11C0. 0818.0039.D580.0800.0000.125B  ;3670 =0 D_K[.6],CALL[WAIT.LOOP]
U 11C1. 0000.003C.0180.0800.0000.10A8  ;3671  NOP
U 10A8. 0000.003D.0180.0800.0000.10E0  ;3672 =00 CALL[NOINTR]
U 10A9. 0000.003D.0180.0800.0000.10AB  ;3673  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 10AA. 0818.0038.0580.0800.0000.134E  ;3674  D_K[.1]     ;SET MME
U 134E. 0000.003C.4980.3C00.0000.134F  ;3675 = ID[TBER0]_D
U 134F. 0810.0038.4980.0928.0000.1350  ;3676  D_RC[5],K[.FF]   ;SET VIRTUAL ADDRESS TO EDGE OF
U 1350. 0019.0014.4980.0800.0200.1351  ;3677  VA_ALU,ALU_D+K[.FF]  ; PAGE
U 1351. 0F00.003C.0980.0800.0084.7352  ;3678  SC_K[.2],D_0   ;SET THE U-TRAP FLAG
U 1352. 0801.001C.0180.0AF8.0000.1353  ;3679  R[0F]_ALU,D_D.ORNOT.MASK
U 1353. 0800.003C.0180.0A00.0000.1354  ;3680  D_R[0]
U 1354. 0000.003C.0180.3000.0000.11C2  ;3681  MCT/WRITE.V.WCHK  ;VIRTUAL REFERENCE
U 11C2. 0818.0039.D580.0800.0000.125B  ;3682 =0 D_K[.6],CALL[WAIT.LOOP]
U 11C3. 0000.003C.0180.0800.0000.10AC  ;3683  NOP
U 10AC. 0000.003D.0180.0800.0000.10E0  ;3684 =00 CALL[NOINTR]
U 10AD. 0000.003D.0180.0800.0000.10AB  ;3685  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 10AE. 0000.003C.0180.0A78.0010.1355  ;3686  ALU_R[0F],CLK.UBCC  ;CHECK THE U-TRAP FLAG
U 1355. 0000.013C.0180.0800.0000.11C4  ;3687 = Z?
U 11C4. 0000.003D.0180.0800.0000.10AB  ;3688 =0 ERROR1     ;NO PAGE BOUNDRY U-TRAP
U 11C5. 0010.0038.0180.0960.0200.1356  ;3689  VA_RC[0C]   ;CHECK MEMORY CONTENTS
U 1356. 0000.003C.0180.C800.0000.1357  ;3690  MCT/READ.P    ;READ LAST LONG WORD OF PAGE
U 1357. 0001.003C.0180.09E8.0000.10B0  ;3691  RC[0D]_D
U 10B0. 0000.003D.0180.0800.0000.10E0  ;3692 =00 CALL[NOINTR]
U 10B1. 0000.003D.0180.0800.0000.10AB  ;3693  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 10B2. 0010.0038.0180.0968.0010.1358  ;3694  ALU_RC[0D],CLK.UBCC  ;TEST
U 1358. 0000.013C.0180.0800.0000.11C6  ;3695 = Z?
U 11C6. 0000.003D.0180.0800.0000.10AB  ;3696 =0 ERROR1     ;LOW(LAST WORD OF PAGE)DATA
          ;3697       ;     WAS CHANGED
U 11C7. 0010.0038.01C0.0960.0000.1359  ;3698  Q_RC[0C]   ;CHECK UPPER LONG WORD I.E.
U 1359. 0019.2014.11C0.09E0.0200.135A  ;3699  RC[0C]_ALU,VA_ALU,Q_Q+K[.4] ;  FIRST WORD OF THE NEXT PAGE
U 135A. 0000.003C.0180.C800.0000.135B  ;3700  MCT/READ.P    ;READ MEMORY
U 135B. 0001.003C.0180.09E8.0000.11C8  ;3701  RC[0D]_D
U 11C8. 0818.0039.D580.0800.0000.125B  ;3702 =0 D_K[.6],CALL[WAIT.LOOP]  ;SAVE DATA READ
U 11C9. 0000.003C.0180.0800.0000.10B4  ;3703  NOP
U 10B4. 0000.003D.0180.0800.0000.10E0  ;3704 =00 CALL[NOINTR]
U 10B5. 0000.003D.0180.0800.0000.10AB  ;3705  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!!
U 10B6. 0010.0038.0180.0968.0010.135C  ;3706  ALU_RC[0D],CLK.UBCC
U 135C. 0000.013C.0180.0800.0000.11CA  ;3707 = Z?
U 11CA. 0000.003D.0180.0800.0000.10AB  ;3708 =0 ERROR1     ;UPPER(FIRST OF NEXT PAGE) LONG
          ;3709       ;   WORD WAS CHANGED
U 11CB. 0000.003C.0180.0800.0000.135D  ;3710  NOP
U 135D. 0000.003C.0180.0800.0000.11CC  ;3711  NOP
          ;3712 ;
          ;3713 ;=============================================================================
          ;3714 ;*
          ;3715 ;SUBTEST C  CANCELLING OF STALL
          ;3716 ;
          ;3717 ;  THIS TEST CHECKS TO SEE THAT STALL CAN
          ;3718 ; BE CANCELLED.  A READ TO A VALID ADDRESS WITH A
          ;3719 ; PARITY ERROR PRESENT WILL BE ATTEMPTED WHEN
          ;3720 ; A TB MISS OCCURES.  THE U-TRAP FLAG
          ;3721 ; WILL BE SET TO RETURN ON EITHER A TIMEOUT OR
```

```
            ;3722 ; TB MISS.  THE RETURNED U-TRAP FLAG SHOULD RECORD
            ;3723 ; THE TB MISS AND NO TIMEOUT.  THE PARITY
            ;3724 ; ERROR(SBI) WILL CAUSE A NO RESPONSE FROM THE
            ;3725 ; CONTROLLER AND ULTIMATELY A TIMEOUT.  IF STALL IS
            ;3726 ; CANCELLED, THEN NO TIMEOUT SHOULD OCCUR.
            ;3727 ;-
            ;3728 ;=============================================================================
            ;3729 ;
U 11CC. 0000.003D.0180.0800.0000.10BB  ;3730 =0 SUBTEST
U 11CD. 0000.003C.0180.0800.0000.11CE  ;3731  NOP
U 11CE. 0000.003D.0180.0800.0000.10A7  ;3732 =0 ERLOOP
U 11CF. 0000.003C.0180.0800.0000.11D0  ;3733  NOP
U 11D0. 0000.003D.0180.0800.0000.1261  ;3734 =0 CALL[INIT]
U 11D1. 0000.003C.0180.0800.0000.11D2  ;3735  NOP
U 11D2. 0000.003D.0180.0800.0000.1277  ;3736 =0 CALL[TBVAL]     ;SET TB VALID
U 11D3. 0F00.003C.2180.0800.0084.735E  ;3737  D_0,SC_K[.14]
U 135E. 0801.001C.0180.0800.0000.135F  ;3738  D_D.ORNOT.MASK     ;SET FORCE REPLACE BOTH
U 135F. 0000.003C.4980.3C00.0000.1360  ;3739  ID[TBER0]_D
U 1360. 0F10.0038.0180.0920.0200.1361  ;3740  VA_RC[04],D_0     ;GET PTE
U 1361. 0000.003C.4180.3C00.0000.1362  ;3741  ID[TBUF]_D     ;WRITE PTE
U 1362. 0818.0038.0580.0800.0000.1363  ;3742  D_K[.1]      ;ENABLE MEMORY MANAGEMENT
U 1363. 0000.003C.4980.3C00.0000.1364  ;3743  ID[TBER0]_D
U 1364. 0F00.003C.D580.0800.0084.7365  ;3744  D_0,SC_K[.6]     ;CATCH TB MISS
U 1365. 0801.001C.0180.0800.0000.1366  ;3745  D_D.ORNOT.MASK
U 1366. 0000.003C.8980.0800.0084.7367  ;3746  SC_K[.D]
U 1367. 0801.001C.0180.0AF8.0000.1368  ;3747  R[0F]_ALU,D_D.ORNOT.MASK  ;CATCH TIMEOUT
U 1368. 0F00.003C.8D80.0800.0084.7369  ;3748  SC_K[.1F],D_0     ;SET FORCE SBI P0 FAULT
U 1369. 0801.001C.0180.0800.0000.136A  ;3749  D_D.ORNOT.MASK
U 136A. 0000.003C.75F0.2C00.0000.136B  ;3750  Q_ID[MAINT]     ;GET CACHE  OFF BITS
U 136B. 081D.0030.0180.0800.0000.136C  ;3751  D_D[OR]Q
U 136C. 0000.003C.7580.3C00.0000.136D  ;3752  ID[MAINT]_D     ;WRITE CACHE OFF, FORCE
            ;3753         ;  P0 FAULT
U 136D. 0000.003C.0180.4000.0000.136E  ;3754  MCT/READ.V.RCHK     ;VIRTUAL REFERENCE
U 136E. 0000.003C.D5C0.0A78.0084.736F  ;3755  SC_K[.6],Q_R[0F]    ;CHECK U-TRAP FLAG
U 136F. 0001.2024.0180.0800.0010.1370  ;3756  ALU_Q.ANDNOT.MASK,CLK.UBCC  ;      FOR TB MISS CLEAR
U 1370. 0000.013C.8980.0800.0084.71D4  ;3757  Z?,SC_K[.D]
U 11D4. 0000.003D.0180.0800.0000.10AB  ;3758 =0 ERROR1      ;DIDN'T TB MISS U-TRAP
U 11D5. 0001.2024.0180.0800.0010.1371  ;3759  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1371. 0000.013C.0180.0800.0000.11D6  ;3760  Z?
U 11D6. 0000.003C.0180.0800.0000.10B8  ;3761 =0 J/C06800     ;NO TIMEOUT
U 11D7. 0000.003D.0180.0800.0000.10AB  ;3762  ERROR1      ;TIMEOUT OCCURED STALL
            ;3763         ;WAS NOT CANCELLED
            ;3764 =00
U 10B8. 0000.003D.0180.0800.0000.10E0  ;3765 C06800: CALL[NOINTR]     ;NO PARITY FAULT(SBI-
            ;3766         ; FAULT-INTERRUPT)
            ;3767         ; SHOULD HAVE OCCURRED
U 10B9. 0000.003D.0180.0800.0000.10AB  ;3768  ERROR1      ;UNEXPECTED INTERRUPT
U 10BA. 0000.003C.0180.0800.0000.11D8  ;3769 C06END: NOP      ; ALL DONE.
            ;3770 =
            ;3771
```

```
            ;3772  .PAGE "TEST 1DA RESERVED"
            ;3773 =0
U 11D8, 0000,003D,0180,0800,0000,104F  ;3774 T1DA: NEWTST
U 11D9, 0000,003C,0180,0800,0000,1372  ;3775  NOP
            ;3776
```

```
            ;3777  .PAGE "DUMMY NEWTST"
U 1372. 0000,003D,0180,0800,0000,104F  ;3778 DUM: NEWTST
            ;3779
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1632
ESKAR4F.MCR                        MICRO2  1P(00)    26-JUL-85  17:03:10        Page  10
:                                Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1907: 1874 1881= 1897= 2524= 2525= 2526= 2528=
U 1008   2247= 2248= 2249= 2250= 2251= 2252= 2253= 2254=
U 1010   2572= 2573= 2574= 2576= 2651= 2652= 2653= 1875
U 1018   2658= 2659= 2660= 1876 2664= 2665= 2666= 1877
U 1020   2670= 2673= 2675= 2676= 3164= 3165= 3166= 1878
U 1028   3174= 3175= 3176= 1879 3179= 3180= 3181= 1880
U 1030   3207= 3208= 3209= 1900 3217= 3218= 3219= 1901
U 1038   3222= 3223= 3224= 1902 3256= 3257= 3258= 1903
U 1040   3267= 3268= 3269= 1904 3276= 3277= 3278= 1905
U 1048   3312= 3313= 3314= 1906 1898= 1899= 2016: 1948
U 1050   3321= 3322= 3323= 1949 3353= 3354= 3355= 1950
U 1058   1959= 1961= 2062: 1951 1972= 1973= 2096: 1952
U 1060   3365= 3366= 3367= 1953 3375= 3376= 3377= 1954
U 1068   3407= 3408= 3409= 1955 3419= 3420= 3421= 1956
U 1070   3427= 3428= 3429= 1957 3459= 3460= 3461= 3462=
U 1078   3471= 3472= 3473= 1958 3479= 3480= 3481= 1962
U 1080   3510= 3511= 3512= 1963 3522= 3523= 3524= 1964
U 1088   3530= 3531= 3532= 1965 3562= 3563= 3564= 1966
U 1090   3573= 3574= 3575= 1967 3581= 3582= 3583= 1968
U 1098   3614= 3615= 3616= 1969 3625= 3626= 3627= 1970
U 10A0   3633= 3634= 3635= 1971 3666= 3667= 3668= 1991
U 10A8   3672= 3673= 3674= 2014 3684= 3685= 3686= 2015
U 10B0   3692= 3693= 3694= 2060 3704= 3705= 3706= 2061
U 10B8   3765= 3768= 3769= 2125 2034= 2035= 2080= 2081=
U 10C0   2226= 2227= 2235= 2236= 2237= 2239= 2242= 2243=
U 10C8   2284= 2285= 2286= 2287= 2294= 2295= 2301= 2303=
U 10D0   2306= 2307= 2326= 2327= 2329= 2330= 2421= 2422=
U 10D8   2425= 2426= 2437= 2438= 2441= 2442= 2479= 2480=
U 10E0   2719= 2720= 2723= 2724= 2726= 2727= 2731= 2732=
U 10E8   2812= 2813= 2836= 2837= 2856= 2857= 2859= 2860=
U 10F0   2862= 2863= 2876= 2877= 2878= 2880= 2881= 2882=
U 10F8   2883= 2884= 2885= 2886= 2887= 2888= 2889= 2890=
U 1100   1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108   1869: 2126 2892= 2893= 1870: 1871: 1872: 1873:
U 1110   3071= 3072= 3073= 3074= 3106= 3107= 3113= 3114=
U 1118   3120= 3121= 3140= 3141= 3153= 3154= 3155= 3156=
U 1120   3157= 3158= 3159= 3160= 3162= 3163= 3172= 3173=
U 1128   3184= 3185= 3195= 3196= 3197= 3198= 3199= 3200=
U 1130   3201= 3202= 3205= 3206= 3215= 3216= 3227= 3228=
U 1138   3238= 3239= 3240= 3241= 3242= 3243= 3244= 3246=
U 1140   3254= 3255= 3265= 3266= 3271= 3273= 3281= 3282=
U 1148   3292= 3293= 3294= 3295= 3296= 3297= 3298= 3300=
U 1150   3310= 3311= 3326= 3327= 2133 2117: 3336= 3337=
U 1158   3338= 3339= 3340= 3341= 3342= 3343= 3351= 3352=
U 1160   3363= 3364= 3369= 3372= 3379= 3380= 3389= 3390=
U 1168   3391= 3392= 3393= 3394= 3395= 3396= 3405= 3406=
U 1170   3417= 3418= 3423= 3424= 3431= 3432= 3441= 3442=
U 1178   3443= 3444= 3445= 3446= 3447= 3448= 3457= 3458=
U 1180   3469= 3470= 3475= 3476= 3483= 3484= 3493= 3494=
U 1188   3495= 3496= 3497= 3498= 3499= 3500= 3508= 3509=
U 1190   3520= 3521= 3526= 3527= 3534= 3535= 3544= 3545=
U 1198   3546= 3547= 3548= 3549= 3550= 3551= 3560= 3561=
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1633
ESKAR4F.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:10          Page  10

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   3571= 3572= 3577= 3578= 3585= 3586= 3595= 3596=
U 11A8   3597= 3598= 3599= 3600= 3601= 3603= 3612= 3613=
U 11B0   3623= 3624= 3629= 3630= 3637= 3638= 3647= 3648=
U 11B8   3649= 3650= 3651= 3652= 3653= 3654= 3664= 3665=
U 11C0   3670= 3671= 3682= 3683= 3688= 3689= 3696= 3698=
U 11C8   3702= 3703= 3708= 3710= 3730= 3731= 3732= 3733=
U 11D0   3734= 3735= 3736= 3737= 3758= 3759= 3761= 3762=
U 11D8   3774= 3775= 2134 2140 2141 2142 2165 2185
U 11E0   2186 2187 2188 2228 2229 2230 2231 2240
U 11E8   2241 2245 2246 2259 2260 2262 2263 2264
U 11F0   2266 2271 2272 2273 2275 2280 2282 2283
U 11F8   2291 2292 2296 2297 2298 2299 2300 2308
U 1200   2309 2310 2312 2313 2314 2315 2316 2317
U 1208   2322 2324 2325 2328 2353 2354 2355 2356
U 1210   2385 2386 2388 2389 2390 2417 2419 2420
U 1218   2424 2428 2429 2430 2431 2433 2434 2435
U 1220   2436 2439 2440 2468 2469 2470 2471 2472
U 1228   2473 2474 2475 2477 2478 2603 2604 2605
U 1230   2649 2650 2677 2692 2693 2694 2695 2696
U 1238   2721 2722 2728 2729 2730 2733 2734 2735
U 1240   2736 2737 2738 2739 2740 2757 2758 2759
U 1248   2760 2761 2762 2763 2764 2765 2766 2767
U 1250   2768 2769 2770 2771 2772 2774 2775 2776
U 1258   2777 2778 2782 2807 2838 2839 2840 2841
U 1260   2854 2875 2894 2895 2905 2906 2907 2915
U 1268   2916 2917 2918 2926 2927 2928 2929 2937
U 1270   2938 2939 2940 2947 2948 2949 2950 3085
U 1278   3086 3087 3088 3089 3090 3099 3100 3101
U 1280   3102 3103 3104 3105 3108 3109 3110 3111
U 1288   3112 3115 3116 3117 3118 3119 3122 3123
U 1290   3124 3125 3126 3127 3128 3129 3130 3131
U 1298   3132 3133 3134 3135 3136 3137 3138 3139
U 12A0   3142 3143 3161 3167 3168 3169 3170 3171
U 12A8   3177 3178 2118: 3182 3183 3203 3204 3210
U 12B0   3211 3212 3213 3214 3220 3221 3225 3226
U 12B8   3247 3248 3249 3250 3251 3252 3253 3259
U 12C0   3260 3261 3262 3263 3264 3270 3274 3275
U 12C8   3279 3280 3302 3304 3306 3307 3308 3309
U 12D0   3315 3316 3317 3318 3319 3320 3324 3325
U 12D8   3344 3345 3346 3347 3348 3349 3350 3356
U 12E0   3357 3358 3359 3360 3361 3362 3368 3373
U 12E8   3374 3378 3397 3398 3399 3400 3401 3402
U 12F0   3403 3404 3410 3411 3412 3413 3414 3415
U 12F8   3416 3422 3425 3426 3430 3449 3450 3451
U 1300   3452 3453 3454 3455 3456 3463 3464 3465
U 1308   3466 3467 3468 3474 3477 3478 3482 3501
U 1310   3502 3503 3504 3505 3506 3507 3513 3514
U 1318   3515 3516 3517 3518 3519 3525 3528 3529
U 1320   3533 3553 3554 3555 3556 3557 3558 3559
U 1328   3565 3566 3567 3568 3569 3570 3576 3579
U 1330   3580 3584 3604 3605 3606 3607 3608 3609
U 1338   3610 3611 3617 3618 3619 3620 3621 3622
U 1340   3628 3631 3632 3636 3655 3656 3657 3658
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1634
ESKAR4F.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:10          Page  10

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3659 3660 3661 3662 3663 3669 3675 3676
U 1350  3677 3678 3679 3680 3681 3687 3690 3691
U 1358  3695 3699 3700 3701 3707 3711 3738 3739
U 1360  3740 3741 3742 3743 3744 3745 3746 3747
U 1368  3748 3749 3750 3751 3752 3754 3755 3756
U 1370  3757 3760 3778
U 1378 - 13EF Unused
U 13F0  2101: 2102: 2103: 2104: 2105: 2106: 2107: 2108:
U 13F8  2109: 2110: 2111: 2112: 2113: 2114: 2115: 2116:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR4F	899
Used	899
Remaining	

Total microwords used in memory U: 899
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR4F.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

```
ZZ-ESKAR-V22  Table of Contents                                                SEQ 1637
   ESKAR50.MCR               MICRO2  1P(00)    30-JUL-85  09

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1852  MICROTRAP HANDLER
; 1943  Micro Monitor Interface Routines
; 1944     Newtest Routine
; 2039     Disable Cache Routine
; 2452     Set Trap Mask
; 2481     Initialize the SBI
; 2509     Find MS780-E Memory
; 2661     Generate IO Address
; 2687     Set Starting Address
; 2721     CONFIGURE MS780-E/H
; 2799     Select Controller
; 2838     ENABLE NEXT MS780-E
; 2915     WAIT REFRESH SUBROUTINE
; 2979     Wait Loop Routine
; 3011 TEST 1DB INSTRUCTION BUFFER READ MISS
; 3115 TEST 1DC INSTRUCTION BUFFER HIT & CACHE DATA PARITY ERROR
; 3189 TEST 1DD INSTRUCTION BUFFER TIMEOUT
; 3295 TEST 1DE INSTRUCTION BUFFER READ HIT AND ADDRESS NOT VALID
; 3360 TEST 1DF INSTRUCTION BUFFER READ MISS AND CACHE PARITY ERROR
; 3427 TEST 1E0 IB READ MISS AND CP READ HIT
; 3485 TEST 1E1 INSTRUCTION BUFFER READ MISS AND CP READ MISS
; 3540 TEST 1E2 INSTRUCTION BUFFER READ MISS AND CP WRITE
; 3590 TEST 1E3 INSTRUCTION BUFFER MISS AND CP VALIDATE
; 3640 TEST 1E4 INSTRUCTION BUFFER MISS AND RDS
; 3777 TEST 1E5 SBI ERROR REGISTER CLEAR WITH IB FLUSH
; 3866 TEST 1E6 TIMEOUT ADDRESS REGISTER BITS <31:28>
; 3963 TEST 1E7 RESERVED
; 3967 TEST 1E8 RESERVED
; 3972 DUMMY NEWTST
```

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR50
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; 2.1 Joe Zagarella 29-jul-1985
;1836 ;
;1837 ;  In subroutine "configure ms780-e/h" the call to "select.cntrlr"
;1838 ;  does a write/read to config reg. A, which does not work
;1839 ;  because the SBI at that point is disabled. The insertion
;1840 ;  of a call to "disable.cache" just before the write/read
;1841 ;  enables the SBI and allows the write/read to take place.
;1842 ;
;1843 ;  Also, in subroutine "configure ms780-e/h" I added code at
;1844 ;  label cnfg.try.upper to check for an upper controller in case
;1845 ;  it could not find a lower controller
;1846 ;
;1847 ;
;1848 ;*****************************************************************************
;1849
```

```
            ;1850  .PAGE
            ;1851
            ;1852  .TOC " MICROTRAP HANDLER"
            ;1853 ;*
            ;1854 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1855 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
            ;1856 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1857 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1858 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1859 ;
            ;1860 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1861 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1862 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1863 ; R[0F] AND DO A RETURN0.
            ;1864 ;
            ;1865 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1866 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1867 ; TO THE CONSOLE.
            ;1868 ;
            ;1869 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
            ;1870 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
            ;1871 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
            ;1872 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
            ;1873 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
            ;1874 ;-
            ;1875
U 1100, 0000,003C,19C0,0A78,0084,7001  ;1876 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101, 0000,003C,05C0,0A78,0084,7001  ;1877 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000,003C,09C0,0A78,0084,7001  ;1878 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103, 0000,003C,0DC0,0A78,0084,7001  ;1879 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104, 0000,003C,11C0,0A78,0084,7001  ;1880 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105, 0000,003C,D5C0,0A78,0084,7001  ;1881 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106, 0000,003C,D9C0,0A78,0084,7001  ;1882 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000,003C,5DC0,0A78,0084,7001  ;1883 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108, 0000,003C,01C0,0A78,0084,7001  ;1884 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C, 0000,003C,85C0,0A78,0084,7001  ;1885 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D, 0000,003C,89C0,0A78,0084,7001  ;1886 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E, 0000,003C,65C0,0A78,0084,7001  ;1887 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F, 0000,003C,61C0,0A78,0084,712F  ;1888 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
            ;1889
U 1001, 0001,2024,0180,0800,0010,1007  ;1890 TRPH0: ALU_Q.ANDNOT.MASK,CLK_UBCC ; WAS TRAP EXPECTED?
U 1007, 0000,013C,0180,0800,0000,1002  ;1891  Z?    ; BRANCH IF NO
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1892 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1893
            ;1894
            ;1895 ;*
            ;1896 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1897 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1898 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1899 ;
            ;1900 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1901 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1902 ;       FAULT STATUS REGISTER
            ;1903 ;       SBI ERROR REGISTER
            ;1904 ;       TIMEOUT ADDRESS REGISTER
```

```
                ;1905 ;       MAINTENANCE REGISTER
                ;1906 ;       CACHE PARITY REGISTER
                ;1907 ;       TB ERROR REGISTER 1
                ;1908 ;       TB ERROR REGISTER 0
                ;1909 ;-
                ;1910
U 1003, 0018.0038.1D80.09E8.0000.101C  ;1911 TRPH1: RC[0D]_SC
U 101C, 0000.003D.F580.0800.0084.714B  ;1912 =0 SETRCF[.A]
U 101D, 0000.003C.49F0.2C00.0000.1013  ;1913  Q_ID[TBER0]
U 1013, 0001.203C.4DF0.2DB0.0000.1017  ;1914  RC[6]_Q,Q_ID[TBER1]
U 1017, 0001.203C.65F0.2DB8.0000.104F  ;1915  RC[7]_Q,Q_ID[SBI.ERR]
U 104F, 0001.203C.6DF0.2DD8.0000.105B  ;1916  RC[0B]_Q,Q_ID[FAULT]
U 105B, 0001.203C.75F0.2DE0.0000.105F  ;1917  RC[0C]_Q,Q_ID[MAINT]
U 105F, 0001.203C.79F0.2DC8.0000.1109  ;1918  RC[9]_Q,Q_ID[PARITY]
U 1109, 0001.203C.69F0.2DC0.0000.112E  ;1919  RC[8]_Q,Q_ID[TIME.ADDR]
U 112E, 0001.203C.81F0.2DD0.0000.1000  ;1920  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1147  ;1921 1000: RC[0E]_Q,ERROR1
                ;1922
                ;1923 ;*
                ;1924 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
                ;1925 ;-
                ;1926
U 112F, 0001.2024.0180.0800.0010.1130  ;1927 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 1130, 0000.013C.0180.0800.0000.101E  ;1928  Z?     ; BRANCH IF NO
U 101E, 0001.2036.0180.0AF8.0000.0001  ;1929 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1 ; RETURN
U 101F, 0000.003C.0180.0800.0000.1003  ;1930  J/TRPH1     ; REPORT UNEXPECTED TRAP
                ;1931
                ;1932
                ;1933 ;*
                ;1934 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1935 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1936 ;  NEWTST
                ;1937 ;  ERLOOP
                ;1938 ;  ERROR1
                ;1939 ;  ERROR2
                ;1940 ;
                ;1941
```

```
                ;1942  .PAGE
                ;1943  .TOC " Micro Monitor Interface Routines"
                ;1944  .TOC "    Newtest Routine"
                ;1945 ;++
                ;1946 ; FUNCTIONAL DESCRIPTION:
                ;1947 ;
                ;1948 ; This routine Initializes the following registers:
                ;1949 ;  PSL    to 1F
                ;1950 ;  CES    to 0
                ;1951 ;  ACC.CS    to disable accellerator
                ;1952 ;  SIR    to 0
                ;1953 ;  CLK.CS    to stop interval timer
                ;1954 ;  TBER0    to disable the TB
                ;1955 ;  SBI.MAINT to     0
                ;1956 ;  SBI.ERR    to 0
                ;1957 ;  FAULT    to 0
                ;1958 ;  COMP    to 0
                ;1959 ;  RC[0E]    to  0
                ;1960 ;  R[0F]    to 0
                ;1961 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1962 ; A SCOPE call is then made to the Monitor.
                ;1963 ;
                ;1964 ; CALLING CONSTRAINT:
                ;1965 ;
                ;1966 ; =0
                ;1967 ;
                ;1968 ; SIDE EFFECTS:
                ;1969 ;
                ;1970 ; D Reg is destroyed
                ;1971 ; SC is destroyed
                ;1972 ;--
U 1131. 0000.003C.65F8.0800.0084.7132  ;1973 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1132. 0818.0038.8D80.0800.0000.1133  ;1974  d_k[.1f]  ; D=1F
U 1133. 0D00.003C.0180.0800.0000.1134  ;1975  d_dal.sc  ; D=001F0000
U 1134. 0000.003C.3D80.3C00.0000.1135  ;1976  id[psl]_d  ; psl = 001F0000
U 1135. 0818.0038.0580.0800.0000.1136  ;1977  d_k[.1]  ; Clear the CES register
U 1136. 0D00.003C.0180.0800.0000.1137  ;1978  d_dal.sc  ; D = 00010000
U 1137. 0F00.003C.3180.3C00.0000.1138  ;1979  id[ces]_d,d_0  ; ces = 00010000
U 1138. 0000.003C.5D80.3C00.0000.1139  ;1980  id[acc.cs]_d  ; acc.cs = 0
U 1139. 0000.003C.3980.3C00.0000.113A  ;1981  id[sir]_d  ; sir = 0
U 113A. 0000.003C.2980.3C00.0000.113B  ;1982  id[clk.cs]_d  ; clk.cs = 0
U 113B. 0000.003C.4980.3C00.0000.1020  ;1983  id[tber0]_d  ; tber0 = 0
U 1020. 0000.003D.0180.0800.0000.115F  ;1984 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1985  intrpt.strobe,  ; Strobe interrupts
U 1021. 0818.0038.C180.0800.4000.113C  ;1986      d_k[.ffff]  ; Setup to clear SBI error register and
U 113C. 0801.0028.6580.3C00.0000.113D  ;1987  id[sbi.err]_d,d_not.d ; and fault register
U 113D. 0F00.003C.6D80.3C00.0000.113E  ;1988  id[fault]_d,d_0  ; ...
U 113E. 0000.003C.6D80.3C00.0000.113F  ;1989  id[fault]_d  ; fault = 0
U 113F. 0000.003C.7580.3C00.0000.1140  ;1990  id[maint]_d  ; MAINT = 0
U 1140. 0000.003C.6580.3C00.0000.1141  ;1991  id[sbi.err]_d  ; sbi error reg = 0
U 1141. 0000.003C.7180.3C00.0000.1142  ;1992  id[comp]_d  ; comp reg = 0
U 1142. 0000.003C.E580.3C00.0000.1143  ;1993  id[T9]_d  ; Clear code for subroutine START.TEST
U 1143. 0001.003C.0180.09F0.0000.1144  ;1994  rc[0e]_d  ;
U 1144. 0001.003C.0180.0AF8.0000.1145  ;1995  r[0f]_d  ; Clear expected trap mask
U 1145. 0001.003C.0180.09F8.0000.1022  ;1996  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1022, 0000,003D,0180,0800,0000,114D  ;1997 =0 call[disable.cache] ;
U 1023, 0F03,003C,0180,09E8,0000,105E  ;1998  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1999
U 1146, 0818,0038,0980,0800,0000,105E  ;2000 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 1147, 0810,0038,0180,0978,0000,1148  ;2001 ERROR1: D_RC[0F]
U 1148, 0819,0034,4D80,0800,0000,104E  ;2002  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2003 104E: D_D.OR.K[.4],J/CALLCON
          ;2004
U 1149, 0810,0038,0180,0978,0000,114A  ;2005 ERROR2: D_RC[0F]
U 114A, 0819,0034,4D80,0800,0000,105A  ;2006  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2007 105A: D_D.OR.K[.6],J/CALLCON
          ;2008
          ;2009 105E:
          ;2010 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2011  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
          ;2012
          ;2013 SETRCF:
U 114B, 0010,0038,01C0,0978,0000,114C  ;2014  Q_RC[0F]
          ;2015  RC[0F]_Q.OR.SC,   ; Set Argumnet count to console
U 114C, 0019,2032,1D80,09F8,0000,0001  ;2016   RETURN1   ; ...and return
          ;2017 ;*
          ;2018 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
          ;2019 ;-
          ;2020
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2021 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2022 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2023 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2024 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2025 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2026 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2027 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2028 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2029 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2030 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2031 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2032 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2033 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2034 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2035 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,114D  ;2036 13FF: NOP
          ;2037
```

```
            ;2038  .PAGE
            ;2039  .TOC "    Disable Cache Routine"
            ;2040 ;**
            ;2041 ; FUNCTIONAL DESCRIPTION:
            ;2042 ;
            ;2043 ; This routine disables cache hits by setting bits <16:15>
            ;2044 ; in the maintenance register.
            ;2045 ;
            ;2046 ; CALLING SEQUENCE:
            ;2047 ;
            ;2048 ; =0
            ;2049 ;
            ;2050 ; SIDE EFFECTS:
            ;2051 ;
            ;2052 ; D & Q Registers destroyed
            ;2053 ;--
            ;2054 DISABLE.CACHE:
U 114D, 0818.0038.4580.0800.0000.114E  ;2055  d_k[.8000]  ; ... and seed constant
U 114E, 0500.003C.0180.0800.0000.114F  ;2056  d_d.left  ; Generate final constant
U 114F, 0819.0030.4580.0800.0000.1150  ;2057  d_d.or.k[.8000]  ; ... = 00018000
U 1150, 0000.003E.7580.3C00.0000.0001  ;2058  id[maint]_d,return1 ; Disable cache and return
            ;2059
            ;2060 ;*
            ;2061 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2062 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2063 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2064 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2065 ;
            ;2066 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2067 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2068 ;
            ;2069 ;   =0 D_0,CALL,J/DC5
            ;2070 ;    NOP
            ;2071 ;   =0 CALL,J/DC5
            ;2072 ;-
            ;2073
U 1151, 0018.0038.1980.0800.0000.1157  ;2074 DC0: ALU_0(K),J/S00
U 1152, 0018.0038.C180.0800.0000.1157  ;2075 DCC: ALU_K[.FFFF],J/S00
U 1153, 0018.0038.C180.0800.0000.1158  ;2076 DCD: ALU_K[.FFFF],J/S01
U 1154, 0018.0038.C180.0800.0000.1159  ;2077 DCE: ALU_K[.FFFF],J/S10
U 1156, 0018.0038.C180.0800.0000.115A  ;2078 DCF: ALU_K[.FFFF],J/S11
            ;2079
U 1157, 0118.0038.1B80.0800.0000.115B  ;2080 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1158, 0118.0038.0B80.0800.0000.115B  ;2081 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 1159, 0118.0038.0780.0800.0000.115B  ;2082 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 115A, 0118.0038.C380.0800.0000.115B  ;2083 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
            ;2084
U 115B, 0100.003E.0380.0800.0000.0001  ;2085 SX: D_D.LEFT2,SI/7,RETURN1  ; EXIT
            ;2086
            ;2087
            ;2088 ;*
            ;2089 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2090 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2091 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2092 ; TYPING IT.
```

```
            ;2093 ;-
            ;2094
U 115C, 0810,0038,4D80,0978,0000,115D  ;2095 SUBTST: D_RC[0F],KMX/.FF00
U 115D, 0019,4016,4D80,09F8,0000,0001  ;2096  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2097
U 1155, 0019,2000,05C0,0800,0000,12AB  ;2098 1155: Q_Q-K[.1],J/12AB
            ;2099 ;+
            ;2100 ; THE FOLLOWING 256 LOCATIONS ARE THE DISPATCH TABLE FOR THE CALL 3'S.
            ;2101 ; THE INSTRUCTION AT LOCATION "DISPAT" LOADS THE Q REGISTER WITH HEX 100
            ;2102 ; AND THEN EXECUTES THE CALL 3. THIS TABLE THEN DECREMENTS THE Q REGISTER.
            ;2103 ; THE LAST INSTRUCTION IN THE TABLE DOES A RETURN1, WHICH RETURNS TO
            ;2104 ; THE TEST THAT MADE THE CALL. THE Q REGISTER HAS THE 8 BIT ADDRESS THAT
            ;2105 ; THE INSTRUCTION BUFFER FORCED WHEN THE RETURN IS MADE.
            ;2106 ;-
            ;2107
U 115E, 0038,003B,41C0,0800,0000,1200  ;2108 DISPAT: Q_ALU.LEFT,ALU_K[.80],SI/ZERO,SUB/SPEC,J/DISPATTBL
            ;2109
            ;2110
            ;2111 1200:
            ;2112 DISPATTBL:
U 1200, 0019,2000,05C0,0800,0000,1201  ;2113 1200: Q_Q-K[.1]
U 1201, 0019,2000,05C0,0800,0000,1202  ;2114 1201: Q_Q-K[.1]
U 1202, 0019,2000,05C0,0800,0000,1203  ;2115 1202: Q_Q-K[.1]
U 1203, 0019,2000,05C0,0800,0000,1204  ;2116 1203: Q_Q-K[.1]
U 1204, 0019,2000,05C0,0800,0000,1205  ;2117 1204: Q_Q-K[.1]
U 1205, 0019,2000,05C0,0800,0000,1206  ;2118 1205: Q_Q-K[.1]
U 1206, 0019,2000,05C0,0800,0000,1207  ;2119 1206: Q_Q-K[.1]
U 1207, 0019,2000,05C0,0800,0000,1208  ;2120 1207: Q_Q-K[.1]
U 1208, 0019,2000,05C0,0800,0000,1209  ;2121 1208: Q_Q-K[.1]
U 1209, 0019,2000,05C0,0800,0000,120A  ;2122 1209: Q_Q-K[.1]
U 120A, 0019,2000,05C0,0800,0000,120B  ;2123 120A: Q_Q-K[.1]
U 120B, 0019,2000,05C0,0800,0000,120C  ;2124 120B: Q_Q-K[.1]
U 120C, 0019,2000,05C0,0800,0000,120D  ;2125 120C: Q_Q-K[.1]
U 120D, 0019,2000,05C0,0800,0000,120E  ;2126 120D: Q_Q-K[.1]
U 120E, 0019,2000,05C0,0800,0000,120F  ;2127 120E: Q_Q-K[.1]
U 120F, 0019,2000,05C0,0800,0000,1210  ;2128 120F: Q_Q-K[.1]
U 1210, 0019,2000,05C0,0800,0000,1211  ;2129 1210: Q_Q-K[.1]
U 1211, 0019,2000,05C0,0800,0000,1212  ;2130 1211: Q_Q-K[.1]
U 1212, 0019,2000,05C0,0800,0000,1213  ;2131 1212: Q_Q-K[.1]
U 1213, 0019,2000,05C0,0800,0000,1214  ;2132 1213: Q_Q-K[.1]
U 1214, 0019,2000,05C0,0800,0000,1215  ;2133 1214: Q_Q-K[.1]
U 1215, 0019,2000,05C0,0800,0000,1216  ;2134 1215: Q_Q-K[.1]
U 1216, 0019,2000,05C0,0800,0000,1217  ;2135 1216: Q_Q-K[.1]
U 1217, 0019,2000,05C0,0800,0000,1218  ;2136 1217: Q_Q-K[.1]
U 1218, 0019,2000,05C0,0800,0000,1219  ;2137 1218: Q_Q-K[.1]
U 1219, 0019,2000,05C0,0800,0000,121A  ;2138 1219: Q_Q-K[.1]
U 121A, 0019,2000,05C0,0800,0000,121B  ;2139 121A: Q_Q-K[.1]
U 121B, 0019,2000,05C0,0800,0000,121C  ;2140 121B: Q_Q-K[.1]
U 121C, 0019,2000,05C0,0800,0000,121D  ;2141 121C: Q_Q-K[.1]
U 121D, 0019,2000,05C0,0800,0000,121E  ;2142 121D: Q_Q-K[.1]
U 121E, 0019,2000,05C0,0800,0000,121F  ;2143 121E: Q_Q-K[.1]
U 121F, 0019,2000,05C0,0800,0000,1220  ;2144 121F: Q_Q-K[.1]
U 1220, 0019,2000,05C0,0800,0000,1221  ;2145 1220: Q_Q-K[.1]
U 1221, 0019,2000,05C0,0800,0000,1222  ;2146 1221: Q_Q-K[.1]
U 1222, 0019,2000,05C0,0800,0000,1223  ;2147 1222: Q_Q-K[.1]
```

```
U 1223. 0019.2000.05C0.0800.0000.1224  ;2148 1223: Q_Q-K[.1]
U 1224. 0019.2000.05C0.0800.0000.1225  ;2149 1224: Q_Q-K[.1]
U 1225. 0019.2000.05C0.0800.0000.1226  ;2150 1225: Q_Q-K[.1]
U 1226. 0019.2000.05C0.0800.0000.1227  ;2151 1226: Q_Q-K[.1]
U 1227. 0019.2000.05C0.0800.0000.1228  ;2152 1227: Q_Q-K[.1]
U 1228. 0019.2000.05C0.0800.0000.1229  ;2153 1228: Q_Q-K[.1]
U 1229. 0019.2000.05C0.0800.0000.122A  ;2154 1229: Q_Q-K[.1]
U 122A. 0019.2000.05C0.0800.0000.122B  ;2155 122A: Q_Q-K[.1]
U 122B. 0019.2000.05C0.0800.0000.122C  ;2156 122B: Q_Q-K[.1]
U 122C. 0019.2000.05C0.0800.0000.122D  ;2157 122C: Q_Q-K[.1]
U 122D. 0019.2000.05C0.0800.0000.122E  ;2158 122D: Q_Q-K[.1]
U 122E. 0019.2000.05C0.0800.0000.122F  ;2159 122E: Q_Q-K[.1]
U 122F. 0019.2000.05C0.0800.0000.1230  ;2160 122F: Q_Q-K[.1]
U 1230. 0019.2000.05C0.0800.0000.1231  ;2161 1230: Q_Q-K[.1]
U 1231. 0019.2000.05C0.0800.0000.1232  ;2162 1231: Q_Q-K[.1]
U 1232. 0019.2000.05C0.0800.0000.1233  ;2163 1232: Q_Q-K[.1]
U 1233. 0019.2000.05C0.0800.0000.1234  ;2164 1233: Q_Q-K[.1]
U 1234. 0019.2000.05C0.0800.0000.1235  ;2165 1234: Q_Q-K[.1]
U 1235. 0019.2000.05C0.0800.0000.1236  ;2166 1235: Q_Q-K[.1]
U 1236. 0019.2000.05C0.0800.0000.1237  ;2167 1236: Q_Q-K[.1]
U 1237. 0019.2000.05C0.0800.0000.1238  ;2168 1237: Q_Q-K[.1]
U 1238. 0019.2000.05C0.0800.0000.1239  ;2169 1238: Q_Q-K[.1]
U 1239. 0019.2000.05C0.0800.0000.123A  ;2170 1239: Q_Q-K[.1]
U 123A. 0019.2000.05C0.0800.0000.123B  ;2171 123A: Q_Q-K[.1]
U 123B. 0019.2000.05C0.0800.0000.123C  ;2172 123B: Q_Q-K[.1]
U 123C. 0019.2000.05C0.0800.0000.123D  ;2173 123C: Q_Q-K[.1]
U 123D. 0019.2000.05C0.0800.0000.123E  ;2174 123D: Q_Q-K[.1]
U 123E. 0019.2000.05C0.0800.0000.123F  ;2175 123E: Q_Q-K[.1]
U 123F. 0019.2000.05C0.0800.0000.1240  ;2176 123F: Q_Q-K[.1]
U 1240. 0019.2000.05C0.0800.0000.1241  ;2177 1240: Q_Q-K[.1]
U 1241. 0019.2000.05C0.0800.0000.1242  ;2178 1241: Q_Q-K[.1]
U 1242. 0019.2000.05C0.0800.0000.1243  ;2179 1242: Q_Q-K[.1]
U 1243. 0019.2000.05C0.0800.0000.1244  ;2180 1243: Q_Q-K[.1]
U 1244. 0019.2000.05C0.0800.0000.1245  ;2181 1244: Q_Q-K[.1]
U 1245. 0019.2000.05C0.0800.0000.1246  ;2182 1245: Q_Q-K[.1]
U 1246. 0019.2000.05C0.0800.0000.1247  ;2183 1246: Q_Q-K[.1]
U 1247. 0019.2000.05C0.0800.0000.1248  ;2184 1247: Q_Q-K[.1]
U 1248. 0019.2000.05C0.0800.0000.1249  ;2185 1248: Q_Q-K[.1]
U 1249. 0019.2000.05C0.0800.0000.124A  ;2186 1249: Q_Q-K[.1]
U 124A. 0019.2000.05C0.0800.0000.124B  ;2187 124A: Q_Q-K[.1]
U 124B. 0019.2000.05C0.0800.0000.124C  ;2188 124B: Q_Q-K[.1]
U 124C. 0019.2000.05C0.0800.0000.124D  ;2189 124C: Q_Q-K[.1]
U 124D. 0019.2000.05C0.0800.0000.124E  ;2190 124D: Q_Q-K[.1]
U 124E. 0019.2000.05C0.0800.0000.124F  ;2191 124E: Q_Q-K[.1]
U 124F. 0019.2000.05C0.0800.0000.1250  ;2192 124F: Q_Q-K[.1]
U 1250. 0019.2000.05C0.0800.0000.1251  ;2193 1250: Q_Q-K[.1]
U 1251. 0019.2000.05C0.0800.0000.1252  ;2194 1251: Q_Q-K[.1]
U 1252. 0019.2000.05C0.0800.0000.1253  ;2195 1252: Q_Q-K[.1]
U 1253. 0019.2000.05C0.0800.0000.1254  ;2196 1253: Q_Q-K[.1]
U 1254. 0019.2000.05C0.0800.0000.1255  ;2197 1254: Q_Q-K[.1]
U 1255. 0019.2000.05C0.0800.0000.1256  ;2198 1255: Q_Q-K[.1]
U 1256. 0019.2000.05C0.0800.0000.1257  ;2199 1256: Q_Q-K[.1]
U 1257. 0019.2000.05C0.0800.0000.1258  ;2200 1257: Q_Q-K[.1]
U 1258. 0019.2000.05C0.0800.0000.1259  ;2201 1258: Q_Q-K[.1]
U 1259. 0019.2000.05C0.0800.0000.125A  ;2202 1259: Q_Q-K[.1]
```

```
U 125A, 0019,2000,05C0,0800,0000,125B  ;2203 125A: Q_Q-K[.1]
U 125B, 0019,2000,05C0,0800,0000,125C  ;2204 125B: Q_Q-K[.1]
U 125C, 0019,2000,05C0,0800,0000,125D  ;2205 125C: Q_Q-K[.1]
U 125D, 0019,2000,05C0,0800,0000,125E  ;2206 125D: Q_Q-K[.1]
U 125E, 0019,2000,05C0,0800,0000,125F  ;2207 125E: Q_Q-K[.1]
U 125F, 0019,2000,05C0,0800,0000,1260  ;2208 125F: Q_Q-K[.1]
U 1260, 0019,2000,05C0,0800,0000,1261  ;2209 1260: Q_Q-K[.1]
U 1261, 0019,2000,05C0,0800,0000,1262  ;2210 1261: Q_Q-K[.1]
U 1262, 0019,2000,05C0,0800,0000,1263  ;2211 1262: Q_Q-K[.1]
U 1263, 0019,2000,05C0,0800,0000,1264  ;2212 1263: Q_Q-K[.1]
U 1264, 0019,2000,05C0,0800,0000,1265  ;2213 1264: Q_Q-K[.1]
U 1265, 0019,2000,05C0,0800,0000,1266  ;2214 1265: Q_Q-K[.1]
U 1266, 0019,2000,05C0,0800,0000,1267  ;2215 1266: Q_Q-K[.1]
U 1267, 0019,2000,05C0,0800,0000,1268  ;2216 1267: Q_Q-K[.1]
U 1268, 0019,2000,05C0,0800,0000,1269  ;2217 1268: Q_Q-K[.1]
U 1269, 0019,2000,05C0,0800,0000,126A  ;2218 1269: Q_Q-K[.1]
U 126A, 0019,2000,05C0,0800,0000,126B  ;2219 126A: Q_Q-K[.1]
U 126B, 0019,2000,05C0,0800,0000,126C  ;2220 126B: Q_Q-K[.1]
U 126C, 0019,2000,05C0,0800,0000,126D  ;2221 126C: Q_Q-K[.1]
U 126D, 0019,2000,05C0,0800,0000,126E  ;2222 126D: Q_Q-K[.1]
U 126E, 0019,2000,05C0,0800,0000,126F  ;2223 126E: Q_Q-K[.1]
U 126F, 0019,2000,05C0,0800,0000,1270  ;2224 126F: Q_Q-K[.1]
U 1270, 0019,2000,05C0,0800,0000,1271  ;2225 1270: Q_Q-K[.1]
U 1271, 0019,2000,05C0,0800,0000,1272  ;2226 1271: Q_Q-K[.1]
U 1272, 0019,2000,05C0,0800,0000,1273  ;2227 1272: Q_Q-K[.1]
U 1273, 0019,2000,05C0,0800,0000,1274  ;2228 1273: Q_Q-K[.1]
U 1274, 0019,2000,05C0,0800,0000,1275  ;2229 1274: Q_Q-K[.1]
U 1275, 0019,2000,05C0,0800,0000,1276  ;2230 1275: Q_Q-K[.1]
U 1276, 0019,2000,05C0,0800,0000,1277  ;2231 1276: Q_Q-K[.1]
U 1277, 0019,2000,05C0,0800,0000,1278  ;2232 1277: Q_Q-K[.1]
U 1278, 0019,2000,05C0,0800,0000,1279  ;2233 1278: Q_Q-K[.1]
U 1279, 0019,2000,05C0,0800,0000,127A  ;2234 1279: Q_Q-K[.1]
U 127A, 0019,2000,05C0,0800,0000,127B  ;2235 127A: Q_Q-K[.1]
U 127B, 0019,2000,05C0,0800,0000,127C  ;2236 127B: Q_Q-K[.1]
U 127C, 0019,2000,05C0,0800,0000,127D  ;2237 127C: Q_Q-K[.1]
U 127D, 0019,2000,05C0,0800,0000,127E  ;2238 127D: Q_Q-K[.1]
U 127E, 0019,2000,05C0,0800,0000,127F  ;2239 127E: Q_Q-K[.1]
U 127F, 0019,2000,05C0,0800,0000,1280  ;2240 127F: Q_Q-K[.1]
U 1280, 0019,2000,05C0,0800,0000,1281  ;2241 1280: Q_Q-K[.1]
U 1281, 0019,2000,05C0,0800,0000,1282  ;2242 1281: Q_Q-K[.1]
U 1282, 0019,2000,05C0,0800,0000,1283  ;2243 1282: Q_Q-K[.1]
U 1283, 0019,2000,05C0,0800,0000,1284  ;2244 1283: Q_Q-K[.1]
U 1284, 0019,2000,05C0,0800,0000,1285  ;2245 1284: Q_Q-K[.1]
U 1285, 0019,2000,05C0,0800,0000,1286  ;2246 1285: Q_Q-K[.1]
U 1286, 0019,2000,05C0,0800,0000,1287  ;2247 1286: Q_Q-K[.1]
U 1287, 0019,2000,05C0,0800,0000,1288  ;2248 1287: Q_Q-K[.1]
U 1288, 0019,2000,05C0,0800,0000,1289  ;2249 1288: Q_Q-K[.1]
U 1289, 0019,2000,05C0,0800,0000,128A  ;2250 1289: Q_Q-K[.1]
U 128A, 0019,2000,05C0,0800,0000,128B  ;2251 128A: Q_Q-K[.1]
U 128B, 0019,2000,05C0,0800,0000,128C  ;2252 128B: Q_Q-K[.1]
U 128C, 0019,2000,05C0,0800,0000,128D  ;2253 128C: Q_Q-K[.1]
U 128D, 0019,2000,05C0,0800,0000,128E  ;2254 128D: Q_Q-K[.1]
U 128E, 0019,2000,05C0,0800,0000,128F  ;2255 128E: Q_Q-K[.1]
U 128F, 0019,2000,05C0,0800,0000,1290  ;2256 128F: Q_Q-K[.1]
U 1290, 0019,2000,05C0,0800,0000,1291  ;2257 1290: Q_Q-K[.1]
```

```
U 1291, 0019,2000,05C0,0800,0000,1292  ;2258 1291: Q_Q-K[.1]
U 1292, 0019,2000,05C0,0800,0000,1293  ;2259 1292: Q_Q-K[.1]
U 1293, 0019,2000,05C0,0800,0000,1294  ;2260 1293: Q_Q-K[.1]
U 1294, 0019,2000,05C0,0800,0000,1295  ;2261 1294: Q_Q-K[.1]
U 1295, 0019,2000,05C0,0800,0000,1296  ;2262 1295: Q_Q-K[.1]
U 1296, 0019,2000,05C0,0800,0000,1297  ;2263 1296: Q_Q-K[.1]
U 1297, 0019,2000,05C0,0800,0000,1298  ;2264 1297: Q_Q-K[.1]
U 1298, 0019,2000,05C0,0800,0000,1299  ;2265 1298: Q_Q-K[.1]
U 1299, 0019,2000,05C0,0800,0000,129A  ;2266 1299: Q_Q-K[.1]
U 129A, 0019,2000,05C0,0800,0000,129B  ;2267 129A: Q_Q-K[.1]
U 129B, 0019,2000,05C0,0800,0000,129C  ;2268 129B: Q_Q-K[.1]
U 129C, 0019,2000,05C0,0800,0000,129D  ;2269 129C: Q_Q-K[.1]
U 129D, 0019,2000,05C0,0800,0000,129E  ;2270 129D: Q_Q-K[.1]
U 129E, 0019,2000,05C0,0800,0000,129F  ;2271 129E: Q_Q-K[.1]
U 129F, 0019,2000,05C0,0800,0000,12A0  ;2272 129F: Q_Q-K[.1]
U 12A0, 0019,2000,05C0,0800,0000,12A1  ;2273 12A0: Q_Q-K[.1]
U 12A1, 0019,2000,05C0,0800,0000,12A2  ;2274 12A1: Q_Q-K[.1]
U 12A2, 0019,2000,05C0,0800,0000,12A3  ;2275 12A2: Q_Q-K[.1]
U 12A3, 0019,2000,05C0,0800,0000,12A4  ;2276 12A3: Q_Q-K[.1]
U 12A4, 0019,2000,05C0,0800,0000,12A5  ;2277 12A4: Q_Q-K[.1]
U 12A5, 0019,2000,05C0,0800,0000,12A6  ;2278 12A5: Q_Q-K[.1]
U 12A6, 0019,2000,05C0,0800,0000,12A7  ;2279 12A6: Q_Q-K[.1]
U 12A7, 0019,2000,05C0,0800,0000,12A8  ;2280 12A7: Q_Q-K[.1]
U 12A8, 0019,2000,05C0,0800,0000,12A9  ;2281 12A8: Q_Q-K[.1]
U 12A9, 0019,2000,05C0,0800,0000,12AA  ;2282 12A9: Q_Q-K[.1]
U 12AA, 0019,2000,05C0,0800,0000,12AB  ;2283 12AA: Q_Q-K[.1]
U 12AB, 0019,2000,05C0,0800,0000,12AC  ;2284 12AB: Q_Q-K[.1]
U 12AC, 0019,2000,05C0,0800,0000,12AD  ;2285 12AC: Q_Q-K[.1]
U 12AD, 0019,2000,05C0,0800,0000,12AE  ;2286 12AD: Q_Q-K[.1]
U 12AE, 0019,2000,05C0,0800,0000,12AF  ;2287 12AE: Q_Q-K[.1]
U 12AF, 0019,2000,05C0,0800,0000,12B0  ;2288 12AF: Q_Q-K[.1]
U 12B0, 0019,2000,05C0,0800,0000,12B1  ;2289 12B0: Q_Q-K[.1]
U 12B1, 0019,2000,05C0,0800,0000,12B2  ;2290 12B1: Q_Q-K[.1]
U 12B2, 0019,2000,05C0,0800,0000,12B3  ;2291 12B2: Q_Q-K[.1]
U 12B3, 0019,2000,05C0,0800,0000,12B4  ;2292 12B3: Q_Q-K[.1]
U 12B4, 0019,2000,05C0,0800,0000,12B5  ;2293 12B4: Q_Q-K[.1]
U 12B5, 0019,2000,05C0,0800,0000,12B6  ;2294 12B5: Q_Q-K[.1]
U 12B6, 0019,2000,05C0,0800,0000,12B7  ;2295 12B6: Q_Q-K[.1]
U 12B7, 0019,2000,05C0,0800,0000,12B8  ;2296 12B7: Q_Q-K[.1]
U 12B8, 0019,2000,05C0,0800,0000,12B9  ;2297 12B8: Q_Q-K[.1]
U 12B9, 0019,2000,05C0,0800,0000,12BA  ;2298 12B9: Q_Q-K[.1]
U 12BA, 0019,2000,05C0,0800,0000,12BB  ;2299 12BA: Q_Q-K[.1]
U 12BB, 0019,2000,05C0,0800,0000,12BC  ;2300 12BB: Q_Q-K[.1]
U 12BC, 0019,2000,05C0,0800,0000,12BD  ;2301 12BC: Q_Q-K[.1]
U 12BD, 0019,2000,05C0,0800,0000,12BE  ;2302 12BD: Q_Q-K[.1]
U 12BE, 0019,2000,05C0,0800,0000,12BF  ;2303 12BE: Q_Q-K[.1]
U 12BF, 0019,2000,05C0,0800,0000,12C0  ;2304 12BF: Q_Q-K[.1]
U 12C0, 0019,2000,05C0,0800,0000,12C1  ;2305 12C0: Q_Q-K[.1]
U 12C1, 0019,2000,05C0,0800,0000,12C2  ;2306 12C1: Q_Q-K[.1]
U 12C2, 0019,2000,05C0,0800,0000,12C3  ;2307 12C2: Q_Q-K[.1]
U 12C3, 0019,2000,05C0,0800,0000,12C4  ;2308 12C3: Q_Q-K[.1]
U 12C4, 0019,2000,05C0,0800,0000,12C5  ;2309 12C4: Q_Q-K[.1]
U 12C5, 0019,2000,05C0,0800,0000,12C6  ;2310 12C5: Q_Q-K[.1]
U 12C6, 0019,2000,05C0,0800,0000,12C7  ;2311 12C6: Q_Q-K[.1]
U 12C7, 0019,2000,05C0,0800,0000,12C8  ;2312 12C7: Q_Q-K[.1]
```

```
U 12C8, 0019,2000,05C0,0800,0000,12C9   ;2313 12C8: Q_Q-K[.1]
U 12C9, 0019,2000,05C0,0800,0000,12CA   ;2314 12C9: Q_Q-K[.1]
U 12CA, 0019,2000,05C0,0800,0000,12CB   ;2315 12CA: Q_Q-K[.1]
U 12CB, 0019,2000,05C0,0800,0000,12CC   ;2316 12CB: Q_Q-K[.1]
U 12CC, 0019,2000,05C0,0800,0000,12CD   ;2317 12CC: Q_Q-K[.1]
U 12CD, 0019,2000,05C0,0800,0000,12CE   ;2318 12CD: Q_Q-K[.1]
U 12CE, 0019,2000,05C0,0800,0000,12CF   ;2319 12CE: Q_Q-K[.1]
U 12CF, 0019,2000,05C0,0800,0000,12D0   ;2320 12CF: Q_Q-K[.1]
U 12D0, 0019,2000,05C0,0800,0000,12D1   ;2321 12D0: Q_Q-K[.1]
U 12D1, 0019,2000,05C0,0800,0000,12D2   ;2322 12D1: Q_Q-K[.1]
U 12D2, 0019,2000,05C0,0800,0000,12D3   ;2323 12D2: Q_Q-K[.1]
U 12D3, 0019,2000,05C0,0800,0000,12D4   ;2324 12D3: Q_Q-K[.1]
U 12D4, 0019,2000,05C0,0800,0000,12D5   ;2325 12D4: Q_Q-K[.1]
U 12D5, 0019,2000,05C0,0800,0000,12D6   ;2326 12D5: Q_Q-K[.1]
U 12D6, 0019,2000,05C0,0800,0000,12D7   ;2327 12D6: Q_Q-K[.1]
U 12D7, 0019,2000,05C0,0800,0000,12D8   ;2328 12D7: Q_Q-K[.1]
U 12D8, 0019,2000,05C0,0800,0000,12D9   ;2329 12D8: Q_Q-K[.1]
U 12D9, 0019,2000,05C0,0800,0000,12DA   ;2330 12D9: Q_Q-K[.1]
U 12DA, 0019,2000,05C0,0800,0000,12DB   ;2331 12DA: Q_Q-K[.1]
U 12DB, 0019,2000,05C0,0800,0000,12DC   ;2332 12DB: Q_Q-K[.1]
U 12DC, 0019,2000,05C0,0800,0000,12DD   ;2333 12DC: Q_Q-K[.1]
U 12DD, 0019,2000,05C0,0800,0000,12DE   ;2334 12DD: Q_Q-K[.1]
U 12DE, 0019,2000,05C0,0800,0000,12DF   ;2335 12DE: Q_Q-K[.1]
U 12DF, 0019,2000,05C0,0800,0000,12E0   ;2336 12DF: Q_Q-K[.1]
U 12E0, 0019,2000,05C0,0800,0000,12E1   ;2337 12E0: Q_Q-K[.1]
U 12E1, 0019,2000,05C0,0800,0000,12E2   ;2338 12E1: Q_Q-K[.1]
U 12E2, 0019,2000,05C0,0800,0000,12E3   ;2339 12E2: Q_Q-K[.1]
U 12E3, 0019,2000,05C0,0800,0000,12E4   ;2340 12E3: Q_Q-K[.1]
U 12E4, 0019,2000,05C0,0800,0000,12E5   ;2341 12E4: Q_Q-K[.1]
U 12E5, 0019,2000,05C0,0800,0000,12E6   ;2342 12E5: Q_Q-K[.1]
U 12E6, 0019,2000,05C0,0800,0000,12E7   ;2343 12E6: Q_Q-K[.1]
U 12E7, 0019,2000,05C0,0800,0000,12E8   ;2344 12E7: Q_Q-K[.1]
U 12E8, 0019,2000,05C0,0800,0000,12E9   ;2345 12E8: Q_Q-K[.1]
U 12E9, 0019,2000,05C0,0800,0000,12EA   ;2346 12E9: Q_Q-K[.1]
U 12EA, 0019,2000,05C0,0800,0000,12EB   ;2347 12EA: Q_Q-K[.1]
U 12EB, 0019,2000,05C0,0800,0000,12EC   ;2348 12EB: Q_Q-K[.1]
U 12EC, 0019,2000,05C0,0800,0000,12ED   ;2349 12EC: Q_Q-K[.1]
U 12ED, 0019,2000,05C0,0800,0000,12EE   ;2350 12ED: Q_Q-K[.1]
U 12EE, 0019,2000,05C0,0800,0000,12EF   ;2351 12EE: Q_Q-K[.1]
U 12EF, 0019,2000,05C0,0800,0000,12F0   ;2352 12EF: Q_Q-K[.1]
U 12F0, 0019,2000,05C0,0800,0000,12F1   ;2353 12F0: Q_Q-K[.1]
U 12F1, 0019,2000,05C0,0800,0000,12F2   ;2354 12F1: Q_Q-K[.1]
U 12F2, 0019,2000,05C0,0800,0000,12F3   ;2355 12F2: Q_Q-K[.1]
U 12F3, 0019,2000,05C0,0800,0000,12F4   ;2356 12F3: Q_Q-K[.1]
U 12F4, 0019,2000,05C0,0800,0000,12F5   ;2357 12F4: Q_Q-K[.1]
U 12F5, 0019,2000,05C0,0800,0000,12F6   ;2358 12F5: Q_Q-K[.1]
U 12F6, 0019,2000,05C0,0800,0000,12F7   ;2359 12F6: Q_Q-K[.1]
U 12F7, 0019,2000,05C0,0800,0000,12F8   ;2360 12F7: Q_Q-K[.1]
U 12F8, 0019,2000,05C0,0800,0000,12F9   ;2361 12F8: Q_Q-K[.1]
U 12F9, 0019,2000,05C0,0800,0000,12FA   ;2362 12F9: Q_Q-K[.1]
U 12FA, 0019,2000,05C0,0800,0000,12FB   ;2363 12FA: Q_Q-K[.1]
U 12FB, 0019,2000,05C0,0800,0000,12FC   ;2364 12FB: Q_Q-K[.1]
U 12FC, 0019,2000,05C0,0800,0000,12FD   ;2365 12FC: Q_Q-K[.1]
U 12FD, 0019,2000,05C0,0800,0000,12FE   ;2366 12FD: Q_Q-K[.1]
U 12FE, 0019,2000,05C0,0800,0000,12FF   ;2367 12FE: Q_Q-K[.1]
```

```
U 12FF, 0019,2002,05C0,0800,0000,0001  ;2368 12FF: Q_Q-K[.1],RETURN1
        ;2369
        ;2370
        ;2371
        ;2372 ;*
        ;2373 ; THIS SUBROUTINE EXECUTE AN SBI UNJAM SEQUENCE
        ;2374 ;-
        ;2375
        ;2376 SBI.UNJAM:
U 115F, 0F00,003C,0180,0800,0000,1160  ;2377 UNJAM: D_0
U 1160, 0000,003C,7580,3C00,0000,1161  ;2378  ID[MAINT]_D   ; RELOAD THE MAT REG
U 1161, 0F00,003C,0180,8000,0000,1162  ;2379  D_0,MCT/SBI.HOLD   ; ASSERT HOLD FOR 16 CYCLES
U 1162, 0019,0020,6180,8000,0010,1163  ;2380 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD ; DONE YET?
U 1163, 0000,013C,0180,8000,0000,1024  ;2381  Z?,MCT/SBI.HOLD   ; BRANCH IF YES
U 1024, 0819,0014,0580,8000,0000,1162  ;2382 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM0 ; CONTINUE HOLDING THE BUS
        ;2383
U 1025, 0F00,003C,0180,8800,0000,1164  ;2384  D_0,MCT/SBI.HOLD+UNJAM  ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
U 1164, 0019,0020,6180,8800,0010,1165  ;2385 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD+UNJAM ; DONE YET?
U 1165, 0000,013C,0180,8800,0000,1026  ;2386  Z?,MCT/SBI.HOLD+UNJAM  ; BRANCH IF YES
U 1026, 0819,0014,0580,8800,0000,1164  ;2387 =0 D_D+K[.1],MCT/SBI.HOLD+UNJAM,J/UNJAM1 ; CONTINUE UNJAM
        ;2388
U 1027, 0F00,003C,0180,8000,8000,1166  ;2389  D_0,MCT/SBI.HOLD,INTRPT.ACK ; ASSERT HOLD FOR 16 CYCLES
U 1166, 0019,0020,6180,8000,0010,1167  ;2390 UNJAM2: ALU_D[XOR]K[.F],MCT/SBI.HOLD,CLK.UBCC ; DONE YET?
U 1167, 0000,013C,8580,8000,0084,7028  ;2391  Z?,MCT/SBI.HOLD,SC_K[.C] ; BRANCH IF YES
U 1028, 0819,0014,0580,8000,0000,1166  ;2392 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM2 ; CONTINUE HOLDING
        ;2393
U 1029, 0803,001C,0180,0800,4000,1168  ;2394  D_NOT.MASK,INTRPT.STROBE
U 1168, 0000,003E,7580,3C00,0000,0001  ;2395  ID[MAINT]_D,RETURN1  ; TURN OFF THE SBI
        ;2396
        ;2397
        ;2398
        ;2399 ;*
        ;2400 ; THE FOLLOWING SUBROUTINE GENERATES CONSTANTS AND SAVES THEM IN THE RA
        ;2401 ; SCRATCH PADS. FOLLOWING ARE THE CONSTANTS GENERATED:
        ;2402 ;
        ;2403 ; R[6] = FORCE REPLACE GROUP 0 AND FORCE MISS GROUP 1
        ;2404 ; R[7] = GROUP 0 AND GROUP 1 MATCH BIT MASK
        ;2405 ; R[8] = GROUP 0 MATCH BIT
        ;2406 ; R[9] = GROUP 1 MATCH BIT
        ;2407 ; R[A] = R[6] PLUS REVERSE PARITY GROUP 0, BYTE 0 (DATA)
        ;2408 ; R[B] = TIMEOUT ADDRESS ( = 01000000 )
        ;2409 ; R[C] = SILO COUNTER MASK
        ;2410 ; R[D] = ADDRESS TO FORCE RDS (4020)
        ;2411 ; R[E] = ADDRESS OF MEMORY CONFIGURATION REGISTER B
        ;2412 ;-
        ;2413
        ;2414 =0
        ;2415 MAINTREGSETUP:
U 102A, 0000,003D,F580,0800,0084,714B  ;2416  SETRCF[.A]   ; Set the Argument count to CONSOLE
U 102B, 0000,003C,0180,0800,0000,102C  ;2417  NOP
U 102C, 0000,003D,0180,0800,0000,11B8  ;2418 =0 CALL[CONFIG.MS780E]  ; Find MS780E memory and configure it
U 102D, 0810,0038,0180,0970,0000,1169  ;2419  D_RC[0E]   ; Get CNFG Reg A Address
U 1169, 0019,0014,1180,0AF0,0000,116A  ;2420  R[0E]_D+K[.4]   ; Make it in CNFG B Address
U 116A, 0818,0038,8580,0800,0000,116B  ;2421  D_K[.C]    ; Form "FORCE RDS" constant for
        ;2422      ; ...CNFG Reg B in R[0D]
```

```
            ;2423       ; ...First ECC substitute bits for
            ;2424       ; ...data of zero
U 116B. 0000.003C,F580.0800.0084,716C  ;2425  SC_K[.A]    ; And now the Diagnostic Mode bits
            ;2426  D_D.ORNOT.MASK,    ; ...and save them in
U 116C. 0801.001C,0180.0AE8.0000,102E  ;2427   R[0D]_ALU  ; ...R[0D]
U 102E. 0000.003D,0180.0800.0000,115F  ;2428 =0 CALL[SBI.UNJAM]   ;
U 102F. 0000.003C,0180.0800.0000,1030  ;2429  NOP
U 1030. 0818.0039,7580.0800.0000,1151  ;2430 =0 D_K[.20],CALL,J/DC0  ; GENERATE 200(X)
U 1031. 0001.003C,1180.0AC0.0084,716D  ;2431  R[8]_D,SC_K[.4]   ; GROUP 0 MATCH BIT
U 116D. 0501.003C,01E0.0AB8.0000,116E  ;2432  R[7]_D,D_D.LEFT,Q_D  ; PART OF MASK
U 116E. 0001.003C,0180.0AC8.0000,116F  ;2433  R[9]_D     ; GROUP 1 MATCH BIT
U 116F. 001D.2030,01F8.0AB8.0000,1170  ;2434  R[7]_Q.OR.D,Q_0   ; COMPLETE THE MASK
U 1170. 0D00.003C,0180.0800.0000,1171  ;2435  D_DAL.SC
U 1171. 0500.003C,65E0.0800.0084,7172  ;2436  Q_D,D_D.LEFT,SC_K[.10]  ; ...
U 1172. 081D.0030,0180.0AB0.0000,1173  ;2437  R[6]_D.OR.Q,D_ALU  ; SAVE
U 1173. 0000.003C,7580.3C00.0000,1174  ;2438  ID[MAINT]_D   ; SETUP THE MAINTENANCE REGISTER
U 1174. 0818.0038,51F8.0800.0000,1175  ;2439  D_K[.1E],Q_0   ; GENERATE CONSTANT FOR R[A]
U 1175. 0D00.003C,01C0.0A30.0000,1176  ;2440  D_DAL.SC,Q_R[6]   ; ...
U 1176. 001D.0030,0180.0AD0.0000,1177  ;2441  R[0A]_D.OR.Q   ; SAVE
U 1177. 0818.0038,6580.0800.0000,1178  ;2442  D_K[.10]   ; GENERATE A TIMEOUT ADDRESS
U 1178. 0800.003C,0180.0800.0000,1179  ;2443  D_D.SWAP   ; ...
U 1179. 0001.003C,0180.0AD8.0000,117A  ;2444  R[0B]_D    ; SAVE
U 117A. 0818.0038,61F8.0800.0000,117B  ;2445  D_K[.F],Q_0
U 117B. 0D00.003C,0180.0800.0000,117C  ;2446  D_DAL.SC
U 117C. 0001.003C,0180.0AE0.0084,717D  ;2447  R[0C]_D,SC_K[.8]  ; SILO COUNTER MASK
U 117D. 0000.003E,0180.0800.0000,0001  ;2448  RETURN1    ; Return to caller
            ;2449
            ;2450
```

```
            ;2451  .PAGE
            ;2452  .TOC "   Set Trap Mask"
            ;2453 ;++
            ;2454 ; FUNCTIONAL DESCRIPTION:
            ;2455 ;
            ;2456 ; This routine is used to set a bit in the Micro Trap Mask
            ;2457 ; R[0F].
            ;2458 ;
            ;2459 ; CALLING CONSTRAINT:
            ;2460 ;
            ;2461 ; =0
            ;2462 ;
            ;2463 ; INPUTS:
            ;2464 ;
            ;2465 ; SC - Contains bit number to set.
            ;2466 ;
            ;2467 ; OUTPUTS:
            ;2468 ;
            ;2469 ; rc[0E] - Contains the current trap mask
            ;2470 ;
            ;2471 ; SIDE EFFECTS:
            ;2472 ;
            ;2473 ; d regsiter is destroyed
            ;2474 ;--
            ;2475 SET_TRAP_MASK:
U 117E, 0800,003C,0180,0A78,0000,117F  ;2476  d_r[0f]
U 117F, 0801,001C,0180,0AF8,0000,1180  ;2477  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1180, 0001,003E,0180,0AF0,0000,0001  ;2478  r[0E]_d,return1    ; Save mask and return
            ;2479
```

```
            ;2480  .PAGE
            ;2481  .TOC "   Initialize the SBI"
            ;2482 ;**
            ;2483 ; FUNCTIONAL DESCRIPTION:
            ;2484 ;
            ;2485 ; This routine performs the following functions:
            ;2486 ;
            ;2487 ;  Executes an SBI Unjam
            ;2488 ;  Clears the SBI Fault Latch
            ;2489 ;  Clears the SBI Error Register
            ;2490 ;
            ;2491 ; CALLING CONSTRAINT:
            ;2492 ;
            ;2493 ; =0
            ;2494 ;
            ;2495 ; SIDE EFFECTS:
            ;2496 ;
            ;2497 ; D & Q Register destroyed
            ;2498 ;--
            ;2499 =0
            ;2500 SBI.INIT:
U 1032, 0000.003D.0180.0800.0000.115F  ;2501  call[sbi.unjam]  ; Unjam the SBI
U 1033, 0818.0038.C180.0800.0000.1181  ;2502  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1181, 0801.0028.6580.3C00.0000.1182  ;2503  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1182, 0F00.003C.6D80.3C00.0000.1183  ;2504  id[fault]_d,d_0
U 1183, 0000.003C.6D80.3C00.0000.1184  ;2505  id[fault]_d
U 1184, 0000.003E.6580.3C00.0000.0001  ;2506  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2507
```

```
            ;2508  .PAGE
            ;2509  .TOC "    Find MS780-E Memory"
            ;2510 ;**
            ;2511 ; FUNCTIONAL DESCRIPTION:
            ;2512 ;
            ;2513 ;          The diagnostic is designed to handle up to 4 (four)
            ;2514 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2515 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2516 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2517 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2518 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2519 ;     first MS780-E found will have its starting address set to 0
            ;2520 ;     (zero).
            ;2521 ;     All the other MS780-E/H's found will have their starting address
            ;2522 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2523 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2524 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2525 ;     current MS780-E unit under test.
            ;2526 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
            ;2527 ;     set to maximum.
            ;2528 ;
            ;2529 ; CALLING CONSTRAINT:
            ;2530 ;
            ;2531 ; =00
            ;2532 ;
            ;2533 ; OUTPUTS:
            ;2534 ;
            ;2535 ; rc[0e] - Contains address of Configuration Register
            ;2536 ; r[0]  - Contains TR level
            ;2537 ;
            ;2538 ; SIDE EFFECTS:
            ;2539 ;
            ;2540 ; D register is destroyed.
            ;2541 ;--
            ;2542 =0
            ;2543 FIND.MS780E:
U 1034, 0000,003D,0180,0800,0000,1032  ;2544  call[sbi.init]  ; Make sure SBI is enabled
U 1035, 0000,003C,0180,0800,0000,1036  ;2545  NOP
U 1036, 0000,003D,0180,0800,0000,114D  ;2546 =0 call[disable.cache] ; Make sure cache is disabled and SBI
            ;2547      ; ...is turned on
U 1037, 0003,003C,0180,0988,0000,1185  ;2548  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1185, 0818,0038,1980,0800,0000,1186  ;2549  d_k[zero]  ; Clear MS780-E/H counter
U 1186, 0000,003C,F980,3C00,0000,1187  ;2550  id[3e]_d  ; ...
U 1187, 0018,0038,0580,0A80,0000,1188  ;2551  r[0]_k[.1]  ; Init TR counter
U 1188, 0F03,003C,0180,09F0,0000,1038  ;2552  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2553      ; ...CNFG A Reg
            ;2554 =0
            ;2555 FIND.MEM.LOOP:
U 1038, 0800,003D,0180,0A00,0000,11AF  ;2556  d_r[0],call[generate.io]; Generate address of TR level
U 1039, 0001,003C,0180,09F0,0200,103A  ;2557  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 103A, 0000,003D,8980,0800,0084,717E  ;2558 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2559  d[long]_cache.p, ; Read the specified tr level
U 103B, 0000,003C,0180,C970,0000,1189  ;2560   lc_rc[0e]  ; ...
U 1189, 0000,003C,0180,0A78,0010,118A  ;2561  alu_r[0f],clk.ubcc ; Check for no response
U 118A, 0001,013C,0180,0880,0082,103C  ;2562  z?,sc_d,ia_ra[0] ; Branch if no adapter here
```

```
U 103C, 0000,003C,0180,0800,0000,118B  ;2563 =0 j/check.adpt.code ; Check for a memory
U 103D, 0000,003C,0180,0800,0000,11AA  ;2564  j/find.mem.lp1  ; Try next tr
            ;2565 CHECK.ADPT.CODE:
U 118B, 0000,003C,1D80,0800,1404,718C  ;2566  state_sc,via.kmx ; Setup to branch on bits<6:4>
U 118C, 0000,163C,0180,0800,0000,1008  ;2567  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,718D  ;2568 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,718E  ;2569  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11AA  ;2570  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11AA  ;2571  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7193  ;2572  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11AA  ;2573  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1197  ;2574  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1197  ;2575  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2576   ;
            ;2577   ; Found an MS780-A or -C. Set the starting address
            ;2578   ; to maximum.
            ;2579   ;
U 118D, 0818,0038,5D80,0800,0000,118F  ;2580 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 118E, 0818,0038,0580,0800,0000,118F  ;2581 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2582 MS780.COMMON:
U 118F, 0819,0030,7180,0800,0000,1190  ;2583  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1190, 0000,003C,01F8,0803,0080,D191  ;2584  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1191, 0D00,003C,0180,0800,0000,1192  ;2585  d_dal.sc  ; Put data in bits<29:14>
            ;2586  cache.p_d[long], ; Set the starting address to maximum
U 1192, 0000,003C,0180,A800,0000,11AA  ;2587       j/find.mem.lp1 ; and continue TR scan
            ;2588   ;
            ;2589   ; Found an MA780. Set the starting address to maximum
            ;2590   ;
            ;2591 MA780.MAX:
U 1193, 0818,0038,39F8,0803,0000,1194  ;2592  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1194, 0D00,003C,0180,0803,0000,1195  ;2593  d_dal.sc,va_va+4 ; Put in proper position
U 1195, 0000,003C,0180,0803,0000,1196  ;2594  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2595  cache.p_d[long], ; Set starting address to maximum
U 1196, 0000,003C,0180,A800,0000,11AA  ;2596       j/find.mem.lp1
            ;2597 ;
            ;2598 ; Found an MS780E
            ;2599 ;
            ;2600
            ;2601 FOUND.MS780E:
U 1197, 0000,003C,F9F0,2C00,0000,1198  ;2602  q_id[3e]  ; Set up to see how many MS780-E's
            ;2603  alu_q.xor.k[.3], ; Are there Maximum units?
U 1198, 0019,2020,0D80,0800,0010,1199  ;2604   clk.ubcc ; Check condition codes
U 1199, 0000,013C,0180,0800,0000,103E  ;2605  z?   ; Are we at maximum units for system
U 103E, 0000,003C,0180,0800,0000,119A  ;2606 =0 J/ms780e.tst  ; No go find how many units ther are
U 103F, 0818,0038,C180,0800,0000,1040  ;2607  d_k[.ffff]  ; Yes set address to maximum
U 1040, 0000,003D,0180,0800,0000,11B3  ;2608 =0 call[set.start.addr] ; .....
U 1041, 0000,003C,0180,0800,0000,11AA  ;2609  j/find.mem.lp1  ; Go check to see if we are done
            ;2610 ;
            ;2611 MS780E.TST:
            ;2612  alu_rc[01],  ; Check "Found MS780E Flag"
U 119A, 0010,0038,0180,0908,0010,119B  ;2613   clk.ubcc ; Check condition codes
U 119B, 0810,0138,0180,0970,0000,1042  ;2614  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2615 =0 j/not.first.ms780e, ; No
U 1042, 0818,0038,C180,0800,0000,1046  ;2616   d_k[.ffff] ; Set starting address
U 1043, 0001,003C,0180,0988,0000,119C  ;2617  rc[01]_d  ; Yes put base address in rc[01]
```

```
U 119C, 0818,0038,7980,0800,0000,119D  ;2618  d_k[.30]  ; Set up SC to point to first unit
U 119D, 0019,0014,F580,0800,0082,119E  ;2619  alu_d+k[.a],sc_alu ; ...
U 119E, 0810,0038,0180,0908,0000,119F  ;2620  d_rc[01]  ; Put base address of unit in D
U 119F, 0000,003C,0180,3400,0000,11A0  ;2621  id(sc)_d  ; Put base address in ID pointed to by SC
U 11A0, 0F00,003C,0180,0800,0000,1044  ;2622  d_0   ; Prepare to set starting address to Zero
U 1044, 0000,003D,0180,0800,0000,11B3  ;2623 =0 call[set.start.addr] ; Go do it
           ;2624  j/find.mem.lp1,  ; Check to see if we are done
U 1045, 0003,003C,0180,09E8,0000,11AA  ;2625   rc[0d]_0 ; ....
           ;2626 =0
           ;2627 NOT.FIRST.MS780E:
U 1046, 0000,003D,0180,0800,0000,11B3  ;2628  call[set.start.addr] ; Go set starting address to maximum
U 1047, 0000,003C,F9F0,2C00,0000,11A1  ;2629  q_id[3e]  ; Get the number of MS780-Es found
U 11A1, 0819,2014,0580,0800,0000,11A2  ;2630  d_q+k[.1]  ; Increment this counter
U 11A2, 0000,003C,F980,3C00,0000,11A3  ;2631  id[3e]_d  ; Save the counter
U 11A3, 0018,0038,11C0,0800,0000,11A4  ;2632  q_k[.4]   ; Prepare to form pointer to the ID registers
           ;2633     ; ...were the I/O base addresses will be stored
U 11A4, 081D,2000,7980,0800,0000,11A5  ;2634  d_q-d,k[.30]  ; ... adjust pointer
U 11A5, 0819,0014,7980,0800,0000,11A6  ;2635  d_d+k[.30]  ; Point the SC to the ID register
U 11A6, 0000,003C,F580,0800,0000,11A7  ;2636  k[.a]   ; ...to receive I/O base address
U 11A7, 0019,0014,F580,0800,0082,11A8  ;2637  alu_d+k[.a],sc_alu ; ...
U 11A8, 0810,0038,0180,0970,0000,11A9  ;2638  d_rc[0e]  ; Get base address to d
U 11A9, 0000,003C,0180,3400,0000,11AA  ;2639  id(sc)_d  ; Move base address to ID pointed to by SC
           ;2640   ;
           ;2641   ; Try next tr
           ;2642   ;
           ;2643 FIND.MEM.LP1:
U 11AA, 0818,0014,0580,0A80,0000,11AB  ;2644  r[0]_1a+k[.1],d_alu ; Increment TR level
           ;2645  alu_d.and.k[.f],
U 11AB, 0019,0034,6180,0800,0010,11AC  ;2646      clk.ubcc  ; Check if all done
U 11AC, 0000,013C,0180,0800,0000,1048  ;2647  z?   ; Branch if yes
U 1048, 0000,003C,0180,0800,0000,1038  ;2648 =0 j/find.mem.loop  ; Try next TR
U 1049, 0810,0038,0180,0908,0010,11AD  ;2649  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11AD, 0000,013C,0180,0800,0000,104A  ;2650  z?   ; Check condition codes
U 104A, 0000,003C,0180,0800,0000,11AE  ;2651 =0 J/FIND.CLEANUP  ; Yes MS780E was found
U 104B, 0000,003E,0180,0800,0000,0001  ;2652  return1   ; No MS780E found
           ;2653 FIND.CLEANUP:
U 11AE, 0001,003C,0180,09F0,0000,104C  ;2654  rc[0e]_d  ; Save Configuration Register A addr. in RC[0E]
U 104C, 0000,003D,0180,0800,0000,1032  ;2655 =0 CALL[SBI.INIT]  ; Clean-up the SBI after causing timeouts in
           ;2656     ; this routine to search for MS780-E/H's
U 104D, 0000,003E,0180,0800,0000,0002  ;2657  return2   ; Do RETURN2 to let the caller know that a
           ;2658     ; MS780-E was found
           ;2659
```

```
            ;2660  .PAGE
            ;2661  .TOC "   Generate IO Address"
            ;2662 ;++
            ;2663 ; FUNCTIONAL DESCRIPTION:
            ;2664 ;
            ;2665 ; This routine is used to generate an SBI Configuration Register
            ;2666 ; address from a TR level.
            ;2667 ;
            ;2668 ; CALLING CONSTRAINT:
            ;2669 ;
            ;2670 ; =0
            ;2671 ;
            ;2672 ; INPUTS:
            ;2673 ;
            ;2674 ; D - Contains TR level
            ;2675 ;
            ;2676 ; OUTPUTS:
            ;2677 ;
            ;2678 ; D - Contains SBI IO address
            ;2679 ;--
            ;2680 GENERATE.IO:
U 11AF, 0000,003C,89F8,0800,0084,71B0  ;2681  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11B0, 0D00,003C,8D80,0800,0084,71B1  ;2682  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11B1, 0000,003C,0980,0800,0084,B1B2  ;2683  sc_sc-k[.2]  ; insert bit 29
U 11B2, 0801,001E,0180,0800,0000,0001  ;2684  d_d.ornot.mask,return1 ; and return
            ;2685
```

```
            ;2686  .PAGE
            ;2687  .TOC "   Set Starting Address"
            ;2688 ;++
            ;2689 ; FUNCTIONAL DESCRIPTION:
            ;2690 ;
            ;2691 ; This routine is used to set the starting address of the
            ;2692 ; memory to the specified value.
            ;2693 ;
            ;2694 ; CALLING CONSTRAINT:
            ;2695 ;
            ;2696 ; =0
            ;2697 ;
            ;2698 ; INPUTS:
            ;2699 ;
            ;2700 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2701 ;     (B Register Image divided by 2**16)
            ;2702 ; rc[0] - Contains Address of Register A
            ;2703 ;
            ;2704 ; OUTPUTS:
            ;2705 ;
            ;2706 ; d - Contains aligned starting address (B Register Image)
            ;2707 ;
            ;2708 ; SIDE EFFECTS:
            ;2709 ;
            ;2710 ; d,q,va, and sc are destroyed
            ;2711 ;--
            ;2712 SET.START.ADDR:
U 11B3, 0010,0038,6580,0970,0284,71B4  ;2713  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11B4, 0000,003C,01F8,0803,0000,11B5  ;2714  va_va+4,q_0  ; Set address to Register B
            ;2715  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11B5, 0D00,003C,0980,0800,0084,B1B6  ;2716      sc_sc-k[.2]  ; Setup to insert bit 14
U 11B6, 0801,001C,0180,0800,0000,11B7  ;2717  d_d.ornot.mask  ; Insert Write Enable bit
U 11B7, 0000,003E,0180,A800,0000,0001  ;2718  cache.p_d[long],return1 ; Set the starting address and return
            ;2719
```

```
;2720  .PAGE
;2721  .TOC "   CONFIGURE MS780-E/H"
;2722 ;*************************************************************
;2723 ;**
;2724 ;
;2725 ; Functional Description:
;2726 ; ----------------------
;2727 ;
;2728 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2729 ; specifically  check  the  memory, but  make use of it to execute.
;2730 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2731 ; that  if  a  Read/Write operation will take place no false errors
;2732 ; will be logged due to misconfigured memory.
;2733 ;
;2734 ; Calling Constraint: =00
;2735 ; ------------------
;2736 ;
;2737 ;
;2738 ; Inputs:
;2739 ; ------
;2740 ;
;2741 ; RC[0E] - I/O BASE OF MS780-E/H
;2742 ;
;2743 ;
;2744 ; Outputs:
;2745 ; -------
;2746 ;
;2747 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2748 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2749 ;      OR IF NO MS780-E/Hs WERE FOUND
;2750 ;      ON THE SBI
;2751 ;
;2752 ; Side Effects:
;2753 ; ------------
;2754 ;
;2755 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2756 ;
;2757 ;
;2758 ;
;2759 ;--
;2760 ;*************************************************************
;2761 CONFIG.MS780E:
U 11B8, 0818,0038,0D80,0800,0000,11B9  ;2762  D_K[.3]    ; Set up flag for the Fail Chain
U 11B9, 0001,003C,0180,09E8,0000,1004  ;2763  RC[0D]_D    ; ...
U 1004, 0000,003D,0180,0800,0000,1034  ;2764 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1005, 0000,003D,0180,0800,0000,1147  ;2765  ERROR1     ; No MS780-E/H's found on the SBI
U 1006, 0000,003C,0180,0800,0000,1050  ;2766  NOP    ; OK. Found at least one MS780-E/H
;2767 =
U 1050, 0000,003D,0180,0800,0000,114D  ;2768 =0 CALL[DISABLE.CACHE]  ; enable sbi
U 1051, 0000,003C,0180,0800,0000,1010  ;2769  nop
;2770 CONFIG.RETRY:    ; Entry point for the case in which the
;2771       ; ...first MS780-E/H could not be
;2772       ; ...properly configured
;2773 =00 D_K[ZERO],   ;
U 1010, 0818,0039,1980,0800,0000,11BB  ;2774   CALL[SELECT.CNTRLR] ; Try to select the lower controller
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1661
ESKAR50.MCR                    MICRO2  1P(00)    30-JUL-85  09:17:39                   Page   7

U 1011, 0000,003C,0180,0800,0000,1014  ;2775  J/CNFG.TRY.UPPER  ; Could NOT configure the lower
          ;2776       ; ...controller, try upper controller
U 1012, 0000,003E,0180,0800,0000,0001  ;2777  RETURN1    ; OK. Lower Controller is configured
          ;2778 =
          ;2779 =00
          ;2780 CNFG.TRY.UPPER:
U 1014, 0818,0039,0980,0800,0000,11BB  ;2781  D_K[.2],CALL[SELECT.CNTRLR] ; Try to select the upper controller
U 1015, 0000,003C,0180,0800,0000,1018  ;2782  J/CNFG.TRY.NEXT   ; Could not configure either controller
U 1016, 0000,003E,0180,0800,0000,0001  ;2783  RETURN1    ; Upper controller is configured
          ;2784 =
          ;2785 =00
          ;2786 CNFG.TRY.NEXT:
U 1018, 0000,003D,0180,0800,0000,11C5  ;2787  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
          ;2788       ; ...MS780-E/H found so go and see
          ;2789       ; ...if there are any more
U 1019, 0000,003C,0180,0800,0000,11BA  ;2790  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
          ;2791       ; ...try to configure it
U 101A, 0818,0038,0D80,0800,0000,101B  ;2792  D_K[.3]    ; Set Flag for Fail Chain
U 101B, 0001,003C,0180,09E8,0000,11BA  ;2793  RC[0D]_D   ; ...
U 11BA, 0000,003D,0180,0800,0000,1147  ;2794  ERROR1    ; Could not configure any MS780-E/H
          ;2795       ; ...abort the test
          ;2796 =
          ;2797
```

```
            ;2798  .PAGE
            ;2799  .TOC "   Select Controller"
            ;2800 ;**
            ;2801 ; FUNCTIONAL DESCRIPTION:
            ;2802 ;
            ;2803 ; This routine is used to put the memory system into the
            ;2804 ; specified configuration with respect to controller select.
            ;2805 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2806 ; a RETURN2 is made.
            ;2807 ;
            ;2808 ; CALLING CONSTRAINT:
            ;2809 ;
            ;2810 ; =00
            ;2811 ;
            ;2812 ; INPUTS:
            ;2813 ;
            ;2814 ; d - Contains value to write to bits<2:0> of Register A
            ;2815 ; rc[0e] - Address of Register A
            ;2816 ;
            ;2817 ; SIDE EFFECTS:
            ;2818 ;
            ;2819 ; sc,d,va are destroyed
            ;2820 ;--
            ;2821 SELECT.CNTRLR:
U 11BB, 0010.0038.0180.0970.0284.71BC  ;2822  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11BC, 0801.001C.0180.0800.0000.11BD  ;2823  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11BD, 0000.003C.0180.A800.0000.11BE  ;2824  cache.p_d[long]  ; Write the configuration Register
U 11BE, 0818.0038.5D80.0800.0000.11BF  ;2825  d_k[.7]  ; Get Misconfiguration bits
U 11BF, 0000.003C.61F8.0800.0084.71C0  ;2826  sc_k[.F],q_0  ; ...
U 11C0, 0D00.003C.0180.0800.0000.11C1  ;2827  d_dal.sc  ; ... D = x38000
U 11C1, 0000.003C.01E0.0800.0000.11C2  ;2828  q_d  ; Save mask
U 11C2, 0000.003C.0180.C800.0000.11C3  ;2829  d[long]_cache.p  ; Read CNFG A Reg and
            ;2830  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11C3, 001D.2034.0180.0800.0010.11C4  ;2831  clk.ubcc ; ...
U 11C4, 0000.013C.0180.0800.0000.1052  ;2832  z?  ; Branch if no
U 1052, 0000.003E.0180.0800.0000.0001  ;2833 =0 RETURN1  ; Bad configuration
U 1053, 0000.003E.0180.0800.0000.0002  ;2834  RETURN2  ; Configuration ok
            ;2835
            ;2836
```

```
            ;2837  .PAGE
            ;2838  .TOC "   ENABLE NEXT MS780-E"
            ;2839 ;**
            ;2840 ; FUNCTIONAL DESCRIPTION:
            ;2841 ;
            ;2842 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2843 ;    They will be tested  separately, according to the TR level at which
            ;2844 ;    they are located,  starting with the one  at the lowest level first.
            ;2845 ;          When  a test has finished  with the first unit, it will have to call
            ;2846 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2847 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2848 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2849 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2850 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2851 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2852 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2853 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2854 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2855 ;
            ;2856 ; CALLING CONSTRAINT:
            ;2857 ;
            ;2858 ;  =00
            ;2859 ;
            ;2860 ;--
            ;2861 ENABLE.NEXT.MS780E:
U 11C5, 0000,003C,F9F0,2C00,0000,11C6  ;2862  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2863  ALU_Q,   ; Check to see if it is zero
U 11C6, 0001,203C,0180,0800,0010,11C7  ;2864     CLK.UBCC  ; Check Condition Codes
U 11C7, 0000,013C,0180,0800,0000,1054  ;2865  Z?   ; ...for zero
U 1054, 0000,003C,0180,0800,0000,11C8  ;2866 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1055, 0000,003E,0180,0800,0000,0002  ;2867  RETURN2   ; Zero No more units to test
            ;2868 ENABLE.NEXT:
U 11C8, 0818,0038,C180,0800,0000,1056  ;2869  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1056, 0000,003D,0180,0800,0000,11B3  ;2870 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1057, 0818,0038,7980,0800,0000,11C9  ;2871  D_K[.30]  ; Prepare to point to the ID register holding
            ;2872     ; ...the I/O Base address of the next MS780-E
U 11C9, 0819,0014,1180,0800,0000,11CA  ;2873  D_D+K[.4]  ; ...
U 11CA, 0000,003C,F580,0800,0000,11CB  ;2874  K[.A]   ; ...
U 11CB, 0819,0014,F580,0800,0000,11CC  ;2875  D_D+K[.A]  ; ...
U 11CC, 0000,003C,F9F0,2C00,0000,11CD  ;2876  Q_ID[3E]  ; Get MS780-E Counter
            ;2877  D_D-Q,   ; D now holds the pointer to the ID register
U 11CD, 081D,0000,0180,0800,0082,11CE  ;2878   SC_ALU  ; ...
U 11CE, 0000,003C,01F0,2400,0000,11CF  ;2879  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11CF, 0001,203C,0180,09F0,0000,11D0  ;2880  RC[0E]_Q  ; Save it also in RC[0E]
U 11D0, 0F03,003C,0180,09E8,0000,1058  ;2881  RC[0D]_0,D_0  ; Set starting address to zero
U 1058, 0000,003D,0180,0800,0000,11B3  ;2882 =0 CALL[SET.START.ADDR] ; ....
U 1059, 0F00,003C,F9F0,2C00,0000,11D1  ;2883  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11D1, 081D,2008,0180,0800,0000,11D2  ;2884  D_Q-D-1   ; Subtract 1 from total
U 11D2, 0000,003C,F980,3C00,0000,11D3  ;2885  ID[3E]_D  ; Adjust the pointer
U 11D3, 0000,003E,0180,0800,0000,0001  ;2886  RETURN1   ; Tell caller another unit was found
            ;2887
            ;2888
            ;2889 ;*
            ;2890 ; THE FOLLOWING ROUTINE WAITS FOR 4 CYCLES FOR THE IB TO GET A DATA WORD
            ;2891 ; THIS ROUTINE MUST BE CALLED WITHIN 3 OR 4 INSTRUCTIONS OF THE "IBC/START"
```

```
            ;2892 ; FUNCTION. IF LESS THAN 3, THE IB WILL NOT GET ANY DATA. IF MORE THAN
            ;2893 ; 4, THE IB WILL GET CACHE DATA ON THE SECOND REFERANCE.
            ;2894 ;-
            ;2895
U 11D4, 0000,003C,0180,F800,0000,11D5  ;2896 WAIT7: MCT/ALLOW.IB.READ
U 11D5, 0000,003C,0180,F800,0000,11D6  ;2897  MCT/ALLOW.IB.READ
U 11D6, 0000,003C,0180,F800,0000,11D7  ;2898  MCT/ALLOW.IB.READ
U 11D7, 1000,003E,0180,F800,0000,0001  ;2899  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2900
            ;2901
            ;2902 ;*
            ;2903 ; THE FOLLOWING ROUTINE WAITS FOR AN IB REFERANCE TO TIMEOUT
            ;2904 ;-
            ;2905
            ;2906 WAITFORTIMEOUT:
U 11D8, 0058,0038,81C0,F800,0000,11D9  ;2907  MCT/ALLOW.IB.READ,Q_K[.3FF].RIGHT  ; SETUP A COUNTER (d512)
U 11D9, 0000,003C,03A8,F800,0000,11DA  ;2908  MCT/ALLOW.IB.READ,Q_Q.LEFT,SI/MUL- ; ...
U 11DA, 0019,2000,05C0,F800,0010,11DB  ;2909 TOWAIT: MCT/ALLOW.IB.READ,Q_Q-K[.1],CLK.UBCC ; CHECK THE COUNTER
U 11DB, 0000,013C,0180,F800,0000,105C  ;2910  MCT/ALLOW.IB.READ,Z?  ; DONE YET?
U 105C, 0000,003C,0180,F800,0000,11DA  ;2911 =0 MCT/ALLOW.IB.READ,J/TOWAIT ; NO
U 105D, 0000,003E,0180,0800,0000,0001  ;2912  RETURN1    ; EXIT
            ;2913
```

```
            ;2914  .PAGE
            ;2915  .TOC "    WAIT REFRESH SUBROUTINE"
            ;2916 ;*************************************************************
            ;2917 ;**
            ;2918 ;
            ;2919 ; Functional Description:
            ;2920 ; ----------------------
            ;2921 ;
            ;2922 ; This subroutine provides the means of synchronizing the
            ;2923 ; diagnostic with the end of a refresh cycle on a MS780-E/H
            ;2924 ; memory.  It should be used by all the tests in which
            ;2925 ; timing is critical and delays due to refresh could
            ;2926 ; introduce false errors.
            ;2927 ;
            ;2928 ; Calling Constraint: =0
            ;2929 ; ------------------
            ;2930 ;
            ;2931 ;
            ;2932 ; Inputs:
            ;2933 ; ------
            ;2934 ;
            ;2935 ; R[0E] must hold the CNFG Reg B Address
            ;2936 ;
            ;2937 ; NOTE: This is slightly different than the similar subroutine
            ;2938 ;  used in ESKAR42 and ESKAR43.  This change is necessary
            ;2939 ;  due to different register assignments in these sections.
            ;2940 ;
            ;2941 ;
            ;2942 ; Outputs:
            ;2943 ; -------
            ;2944 ;
            ;2945 ; RETURN1 at the end of a refresh cycle.
            ;2946 ;
            ;2947 ;
            ;2948 ; Side Effects:
            ;2949 ; ------------
            ;2950 ;
            ;2951 ; The contents of the following registers will be lost:
            ;2952 ;
            ;2953 ; D, SC, CNFG Reg B
            ;2954 ;
            ;2955 ;--
            ;2956 ;*************************************************************
            ;2957 WAITREFRESH:
U 11DC, 0000,003C,0180,0A70,0200,11DD  ;2958  VA_R[0E]  ; Get address of CNFG Reg B
U 11DD, 0000,003C,0180,0800,0084,71DE  ;2959  SC_K[.8]  ; Get data for CNFG  Reg B
U 11DE, 0803,001C,0180,0800,0000,11DF  ;2960  D_NOT.MASK  ; Set bit (Refresh Lock Bit)
U 11DF, 0000,003C,0180,A800,0000,11E0  ;2961  CACHE.P_D[LONG]  ; Set bit 8 in CNFG Reg B
U 11E0, 0F00,003C,0180,0800,0000,11E1  ;2962  D_0  ; Get ready to clear the bit
U 11E1, 0000,003C,0180,A800,0000,11E2  ;2963  CACHE.P_D[LONG]  ; Clear bit 8 in CNFG Reg B
            ;2964 REFRESH.WAIT.LOOP:
            ;2965  D[LONG]_CACHE.P, ; Get contents of CNFG Reg B
U 11E2, 0000,003C,0180,C800,0084,71E3  ;2966  SC_K[.8] ; Point to bit 8
            ;2967  ALU_D.ANDNOT.MASK, ; Is bit 8 still set?
U 11E3, 0001,0024,0180,0800,0010,11E4  ;2968  CLK.UBCC ; ...
```

```
U 11E4, 0000,013C,0180,0800,0000,1060  ;2969  Z?    ;
U 1060, 0000,003C,0180,0800,0000,11E2  ;2970 =0 J/REFRESH.WAIT.LOOP ; Bit 8 is still set
U 1061, 0818,0038,1180,0800,0000,1062  ;2971  D_K[.4]    ; Number of cycles to wait for
          ;2972      ; ...refersh to end
U 1062, 0000,003D,0180,0800,0000,11E5  ;2973 =0 CALL[WAIT.LOOP]  ; Go and wait for refresh to end
U 1063, 0000,003E,0180,0800,0000,0001  ;2974  RETURN1    ; OK. Refresh should be done by now
          ;2975      ; ...Return to caller
          ;2976
          ;2977
```

```
            ;2978  .PAGE
            ;2979  .TOC "   Wait Loop Routine"
            ;2980 ;++
            ;2981 ; FUNCTIONAL DESCRIPTION:
            ;2982 ;
            ;2983 ; This routine is used to wait the specified number of machine cycles.
            ;2984 ; This routine is typically called to wait for an SBI write to
            ;2985 ; I/O space to complete.
            ;2986 ;
            ;2987 ; CALLING CONSTRAINT:
            ;2988 ;
            ;2989 ; =0
            ;2990 ;
            ;2991 ; INPUTS:
            ;2992 ;
            ;2993 ; d - Contains number of cycles to wait
            ;2994 ; alu.z condition code must not be set
            ;2995 ;
            ;2996 ; SIDE EFFECTS:
            ;2997 ;
            ;2998 ; d is destroyed
            ;2999 ;--
            ;3000 WAIT.LOOP:
U 11E5, 0018,0038,6180,0800,0010,1064  ;3001  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;3002      ; ...is not set
            ;3003 =0
            ;3004 W.LOOP:
            ;3005  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1064, 0819,0100,0580,0800,0010,1064  ;3006       j/W.LOOP
U 1065, 0000,003E,0180,0800,0000,0001  ;3007  return1    ; exit
            ;3008
            ;3009
            ;3010
```

```
                ;3011  .PAGE "TEST 1DB INSTRUCTION BUFFER READ MISS"
                ;3012 ;*****************************************************************
                ;3013 ;*
                ;3014 ;
                ;3015 ; INSTRUCTION BUFFER READ MISS
                ;3016 ;
                ;3017 ; TEST DESCRIPTION
                ;3018 ;
                ;3019 ; THIS TEST EXECUTES AN INSTRUCTION BUFFER READ FROM MAIN MEMORY.
                ;3020 ; IT DOES THIS BY INVALIDATING THE CACHE LOCATION BEING REFERANCED,
                ;3021 ; STARTING THE IB, AND ALLOWING THE READ TO COMPLETE (APPROXIMATELY
                ;3022 ; 10 CYCLES).
                ;3023 ;
                ;3024 ; WHILE THE READ IS IN PROGRESS, THE SBI MAINTENANCE REGISTER IS READ AND
                ;3025 ; SAVED SO THAT THE "SBI INTERFACE NOT BUSY" BIT CAN BE CHECKED LATER.
                ;3026 ; WHEN THE CYCLE IS COMPLETE, THE HIT OR MISS BITS ARE CHECKED
                ;3027 ; FOR A CACHE MISS, THE IB DATA IS CHECKED BY EXECUTING A "SUB/SPEC"
                ;3028 ; FUNCTION, AND THE CACHE IS CHECKED TO SEE THAT THE REFERANCE WAS
                ;3029 ; MADE TO BE A HIT.
                ;3030 ;
                ;3031 ; LOGIC DESCRIPTION
                ;3032 ;
                ;3033 ; THIS TEST CHECKS THE LOGIC ON THE SBH AND SBL MODULES THAT CAUSE
                ;3034 ; AN SBI REFERANCE TO BE MADE FROM AN IB REQUEST.
                ;3035 ;
                ;3036 ; ERROR DESCRIPTION
                ;3037 ;
                ;3038 ; THE ERROR TYPEOUT IS KEYED TO THE FOLLOWING LABELS:
                ;3039 ;
                ;3040 ; ERR1: DATA EXPECTED HIT/MISS BITS
                ;3041 ;   RECEIVED HIT/MISS BITS
                ;3042 ; ERR2: DATA EXPECTED "SBI INTERFACE NOT BUSY" BIT
                ;3043 ;   RECEIVED "SBI INTERFACE NOT BUSY" BIT
                ;3044 ; ERR3: DATA EXPECTED UPC WHEN "SUB/SPEC" EXECUTED
                ;3045 ;   RECEIVED UPC WHEN "SUB/SPEC" EXECUTED
                ;3046 ; ERR4: DATA EXPECTED HIT/MISS BITS
                ;3047 ;   RECEIVED HIT/MISS BITS
                ;3048 ;
                ;3049 ; SYNC POINT DESCRIPTION
                ;3050 ; ADDRESS SYNC00--"SBLP RAISE TR FF" SHOULD SET
                ;3051 ;   HIT/MISS BITS ARE LOADED
                ;3052 ; ADDRESS SYNC01--IB READ DATA IS ASCERTED
                ;3053 ;-
                ;3054 ;*****************************************************************
                ;3055
                ;3056 =0
U 1066, 0000.003D.0180.0800.0000.1131  ;3057 T001: NEWTST
U 1067, 0000.003C.0180.0800.0000.1068  ;3058  NOP
U 1068, 0000.003D.0180.0800.0000.102A  ;3059 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 1069, 0000.003C.0180.0800.0000.106A  ;3060  NOP
U 106A, 0818.0039.0180.0800.0000.1156  ;3061 =0 D_K[.8],CALL,J/DCF  ; GENERATE EXPECTED UPC BITS
U 106B, 0001.003C.0180.0A80.0000.11E6  ;3062  R[0]_D    ; SAVE
U 11E6, 0F18.0038.1980.0800.0200.11E7  ;3063 RETRY1: ALU_0(K),VA_ALU,D_0  ; LOAD THE VA
U 11E7, 0000.003C.0180.A800.0000.106C  ;3064  CACHE.P_D[LONG]  ; WRITE DATA INTO MEMORY
U 106C, 0000.003D.0180.0800.0000.1146  ;3065 =0 ERLOOP
```

```
U 106D, 0000.003C.0180.9000.0000.106E  ;3066  CACHE.INVALIDATE  ; INVALIDATE ADDRESS 0 IN THE CACHE
U 106E, 0000.003D.0180.0800.0000.11DC  ;3067 =0 CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 106F, 2018.0038.1980.0800.4200.11E8  ;3068  ALU_0(K),FLUSH.IB  ; LOAD THE VIBA
U 11E8, 3000.003C.0180.6000.0000.11E9  ;3069  LOAD.IB,IBC/START  ; EXECUTE READ.V.NEWPC
U 11E9, 0000.003C.0180.F800.0000.11EA  ;3070 SYNC00: MCT/ALLOW.IB.READ  ; WAIT FOR THE SBI CYCLE TO START
U 11EA, 0000.003C.65F0.2C00.0000.11EB  ;3071  Q_ID[SBI.ERR]  ; READ THE SBI INTERFACE NOT BUSY BIT
U 11EB, 0001.203C.0180.FA88.0000.1070  ;3072  R[1]_Q,MCT/ALLOW.IB.READ ; SAVE THE ERROR REGISTER
U 1070, 0000.003D.0180.F800.0000.11D4  ;3073 =0 CALL,J/WAIT7,MCT/ALLOW.IB.READ ; WAIT FOR THE CYCLE TO COMPLETE
U 1071, 1000.003C.75F0.2C00.0000.11EC  ;3074  IBC/STOP,Q_ID[MAINT]  ; READ THE HIT/MISS BITS
U 11EC, 001C.0034.01C0.0A38.0010.11ED  ;3075  Q_Q.AND.R[7],CLK.UBCC  ; MASK THE HIT/MISS BITS
U 11ED, 0003.013C.0180.09F0.0000.1072  ;3076  Z?,RC[0E]_0  ; BRANCH IF NO HIT
            ;3077 =0
U 1072, 0001.203D.0180.09E8.0000.1149  ;3078 ERR1: ERROR2,RC[0D]_Q  ; CACHE HIT
            ;3079
            ;3080 ;*
            ;3081 ; CHECK THAT THE SBI CYCLE WAS STARTED
            ;3082 ;-
            ;3083
U 1073, 0818.0034.0980.0A08.0010.11EE  ;3084  D_R[1].AND.K[.2],CLK.UBCC ; MASK THE BUSY BIT
U 11EE, 0000.013C.0180.0800.0000.1074  ;3085  Z?  ; BRANCH IF OK
            ;3086 =0
U 1074, 0001.003D.0180.09E8.0000.1149  ;3087 ERR2: ERROR2,RC[0D]_D  ; BUSY BIT DID NOT CLEAR
            ;3088
            ;3089 ;*
            ;3090 ; CHECK THAT THE DATA WAS RECEIVED IN THE IB
            ;3091 ;-
            ;3092
U 1075, 0800.003C.0180.0A00.0000.1076  ;3093  D_R[0]  ; GET EXPECTED UPC
U 1076, 0001.003D.0180.09F0.0000.115E  ;3094 =0 RC[0E]_D,CALL,J/DISPAT  ; EXECUTE THE CALL 3
U 1077, 001D.2000.0180.0800.0010.11EF  ;3095  ALU_Q-D,CLK.UBCC  ; CHECK THE RECEIVED UPC
U 11EF, 0000.013C.0180.0800.0000.1078  ;3096  Z?  ; BRANCH IF OK
            ;3097 =0
U 1078, 0001.203D.0180.09E8.0000.1149  ;3098 ERR3: ERROR2,RC[0D]_Q  ; IB DID NOT GET CORRECT DATA
            ;3099
            ;3100 ;*
            ;3101 ; CHECK THAT THE DATA WAS WRITTEN INTO THE CACHE
            ;3102 ;-
            ;3103
U 1079, 0018.0038.1980.0800.0200.11F0  ;3104  ALU_0(K),VA_ALU  ; LOAD VA WITH ADR OF REFERANCED LOCATION
U 11F0, 0000.003C.0180.C800.0000.11F1  ;3105  D[LONG]_CACHE.P  ; READ THE REFERANCED LOCATION
U 11F1, 0800.003C.75F0.2E40.0000.11F2  ;3106  Q_ID[MAINT],D_R[8]  ; READ THE HIT/MISS BITS
U 11F2, 001C.0034.01C0.0A38.0000.11F3  ;3107  Q_Q.AND.R[7]  ; MASK THEM
U 11F3, 001D.2000.0180.0800.0010.11F4  ;3108  ALU_Q-D,CLK.UBCC  ; CHECK FOR GROUP ZERO HIT
U 11F4, 0001.013C.0180.09F0.0000.107A  ;3109  Z?,RC[0E]_D  ; BRANCH IF YES
            ;3110 =0
U 107A, 0001.203D.0180.09E8.0000.1149  ;3111 ERR4: ERROR2,RC[0D]_Q  ; HIT/MISS BITS INCORRECT
U 107B, 0000.003C.0180.0800.0000.107C  ;3112 END001: NOP
            ;3113
            ;3114
```

```
;3115  .PAGE "TEST 1DC INSTRUCTION BUFFER HIT & CACHE DATA PARITY ERROR"
;3116 ;*****************************************************************
;3117 ;+
;3118 ;
;3119 ; INSTRUCTION BUFFER HIT & CACHE DATA PARITY ERROR
;3120 ;
;3121 ; TEST DESCRIPTION
;3122 ;
;3123 ; THIS TEST EXECUTES AN INSTRUCTION BUFFER READ THAT IS A CACHE HIT
;3124 ; IN GROUP 0 WITH THE REVERSE PARITY ENABLED FOR GROUP 0 BYTE 0.
;3125 ; THE DATA IN THE IB IS THEN TESTED TO ENSURE THAT IT IS STILL INVALID.
;3126 ; THE HIT/MISS BITS ARE CHECKED TO ENSURE THAT THEY DID NOT CHANGE.
;3127 ;
;3128 ; LOGIC DESCRIPTION
;3129 ;
;3130 ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT INHIBITS "IB READ
;3131 ; DATA" WHEN A CACHE PARITY ERROR IS DETECTED ON AN IB HIT.
;3132 ; IT ALSO CHECKS THAT A CACHE PARITY ERROR INHIBITS THE LOAD OF THE
;3133 ; HIT/MISS BITS.
;3134 ;
;3135 ; ERROR DESCRIPTION
;3136 ;
;3137 ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
;3138 ;
;3139 ; ERR5: DATA EXPECTED HIT/MISS BITS
;3140 ;   RECEIVED HIT/MISS BITS
;3141 ; ERR6: DATA EXPECTED UPC WHEN "SUB/SPEC" ASCERTED
;3142 ;   RECEIVED UPC WHEN "SUB/SPEC" ASCERTED
;3143 ;
;3144 ; SYNC POINT DESCRIPTION
;3145 ;
;3146 ; ADDRESS SYNC02--SBLR IB READ DATA SHOULD BE INHIBITED
;3147 ;   SBLP CLOCK HIT REG SHOULD BE INHIBITED
;3148 ;-
;3149 ;*****************************************************************
;3150
;3151 =0
U 107C. 0000.003D.0180.0800.0000.1131  ;3152 T002: NEWTST
U 107D. 0000.003C.0180.0800.0000.107E  ;3153  NOP
U 107E. 0000.003D.0180.0800.0000.102A  ;3154 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 107F. 0000.003C.0180.0800.0000.1080  ;3155  NOP
U 1080. 0818.0039.5D80.0800.0000.1153  ;3156 =0 D_K[.7],CALL,J/DCD  ; GENERATE EXPECTED UPC
U 1081. 0001.003C.0180.0A80.0000.11F5  ;3157  R[0]_D    ; SAVE
U 11F5. 0838.0038.1B80.0800.0200.1082  ;3158  ALU_0(K),VA_ALU,D_ALU.LEFT,SI/7 ; LOAD THE VA,D_"K[.1]"
U 1082. 0000.003D.0180.A800.0000.1146  ;3159 =0 ERLOOP,CACHE.P_D[LONG]  ; WRITE PATTERN TO MEMORY
U 1083. 0818.0038.0580.0800.0000.11F6  ;3160  D_K[.1]
U 11F6. 0000.003C.0180.9000.0000.11F7  ;3161  MCT/INVALIDATE   ; MAKE ADDRESS 0 A MISS
U 11F7. 0000.003C.0180.C800.0000.11F8  ;3162  D[LONG]_CACHE.P   ; INIT THE HIT/MISS BITS
        ;3163       ; MAKES ADDRESS 0 A HIT
U 11F8. 2018.0038.1980.0800.4200.11F9  ;3164  ALU_0(K),FLUSH.IB  ; LOAD THE VIBA
U 11F9. 0800.003C.0180.0A50.0000.11FA  ;3165  D_R[0A]
U 11FA. 0000.003C.7580.3C00.0000.11FB  ;3166  ID[MAINT]_D   ; SET THE INVERT PARITY BITS
U 11FB. 3000.003C.0180.6000.0000.11FC  ;3167  LOAD.IB,IBC/START  ; START THE IB REFERANCE
U 11FC. 0000.003C.0180.F800.0000.11FD  ;3168 SYNC02: MCT/ALLOW.IB.READ  ; WAIT A CYCLE
U 11FD. 1800.003C.0180.FA30.0000.11FE  ;3169  MCT/ALLOW.IB.READ,IBC/STOP,D_R[6] ; STOP THE IB
```

```
U 11FE, 0000,003C,7580,3C00,0000,11FF  ;3170  ID[MAINT]_D   ; CLEAR THE REVERSE PARITY BITS
U 11FF, 0003,003C,75F0,2DF0,0000,1300  ;3171  Q_ID[MAINT],RC[0E]_0  ; READ THE HIT/MISS BITS
U 1300, 001C,0034,01C0,0A38,0010,1301  ;3172  Q_Q.AND.R[7],CLK.UBCC  ; CHECK THAT HIT WAS NOT RECORDED
U 1301, 0000,013C,0180,0800,0000,1084  ;3173  Z?    ; BRANCH IF OK
            ;3174 =0
U 1084, 0001,203D,0180,09E8,0000,1149  ;3175 ERR5: ERROR2,RC[0D]_Q   ; HIT/MISS BITS INCORRECT
            ;3176
            ;3177 ;*
            ;3178 ; NOW CHECK THAT THE IB DID NOT GET THE DATA
            ;3179 ;-
            ;3180
U 1085, 0800,003C,0180,0A00,0000,1086  ;3181  D_R[0]    ; GET EXPECTED UPC
U 1086, 0001,003D,0180,09F0,0000,115E  ;3182 =0 RC[0E]_D,CALL,J/DISPAT  ; EXECUTE THE CALL3
U 1087, 001D,2000,0180,0800,0010,1302  ;3183  ALU_Q-D,CLK.UBCC  ; CHECK THE ADDRESS
U 1302, 0000,013C,0180,0800,0000,1088  ;3184  Z?    ; BRANCH IF OK
            ;3185 =0
U 1088, 0001,203D,0180,09E8,0000,1149  ;3186 ERR6: ERROR2,RC[0D]_Q   ; UPC INCORRECT
U 1089, 0000,003C,0180,0800,0000,108A  ;3187 END002: NOP
            ;3188
```

```
                ;3189  .PAGE "TEST 1DD INSTRUCTION BUFFER TIMEOUT"
                ;3190  ;*****************************************************************
                ;3191  ;+
                ;3192  ;
                ;3193  ; INSTRUCTION BUFFER TIMEOUT
                ;3194  ;
                ;3195  ; TEST DESCRIPTION
                ;3196  ;
                ;3197  ; THIS TEST CHECKS THAT AN INSTRUCTION BUFFER REFERANCE TO NON-EXISTANT
                ;3198  ; MEMORY DOES NOT CAUSE A TIMEOUT MICRO TRAP, DOES NOT CAUSE A TIMEOUT
                ;3199  ; INTERRUPT, THAT THE IB TIMEOUT BIT IN THE ERROR REGISTER SETS AND CLEARS, AND
                ;3200  ; THAT THE IB DOES NOT RECEIVE ANY DATA.
                ;3201  ; THE ADDRESS THAT IS REFERANCED IS 01000000(X).
                ;3202  ;
                ;3203  ; LOGIC DESCRIPTION
                ;3204  ;
                ;3205  ; THIS TEST CHECKS THE LOGIC ON SBL THAT CONTROLS THE TIMEOUT BITS
                ;3206  ; IN THE SBI ERROR REGISTER, THE TIMEOUT MICRO TRAP SIGNAL, AND THE
                ;3207  ; TIMEOUT INTERRUPT SIGNAL.
                ;3208  ;
                ;3209  ; ERROR DESCRIPTION
                ;3210  ;
                ;3211  ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
                ;3212  ;
                ;3213  ; ERR7: DATA EXPECTED MICRO TRAP FLAG
                ;3214  ;   RECEIVED MICRO TRAP FLAG
                ;3215  ; ERR8: DATA EXPECTED INTERRUPT VECTOR
                ;3216  ;   RECEIVED INTERRUPT VECTOR
                ;3217  ; ERR9: DATA EXPECTED SBI ERROR REGISTER
                ;3218  ;   RECEIVED SBI ERROR REGISTER
                ;3219  ; ERRA: DATA EXPECTED SBI ERROR REGISTER
                ;3220  ;   RECEIVED SBI ERROR REGISTER
                ;3221  ; ERRB: DATA EXPECTED UPC WHEN THE "SUB/SPEC" IS ASCERTED
                ;3222  ;   RECEIVED UPC WHEN THE "SUB/SPEC" IS ASCERTED
                ;3223  ;
                ;3224  ;-
                ;3225  ;*****************************************************************
                ;3226
                ;3227  =0
U 108A. 0000.003D.0180.0800.0000.1131   ;3228 T003: NEWTST
U 108B. 0000.003C.0180.0800.0000.108C   ;3229  NOP
U 108C. 0000.003D.0180.0800.0000.102A   ;3230 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 108D. 0000.003C.0180.0800.0000.108E   ;3231  NOP
U 108E. 0818.0039.5D80.0800.0000.1154   ;3232 =0 D_K[.7],CALL,J/DCE  ; GENERATE EXPECTED UPC
U 108F. 0001.003C.0180.0A80.0000.1090   ;3233  R[0]_D    ; SAVE
U 1090. 0000.003D.0180.0800.0000.1146   ;3234 =0 ERLOOP
U 1091. 0F00.003C.8980.0800.0084.7303   ;3235  SC_K[.D],D_0   ; SET THE EXPECTED TIMEOUT
U 1303. 0003.001C.01C0.0AF8.0000.1304   ;3236  R[0F]_NOT.MASK,Q_ALU  ; TRAP FLAG
U 1304. 0001.203C.0180.09F0.0000.1305   ;3237  RC[0E]_Q   ; SET EXPECTED DATA
U 1305. 2000.003C.0180.0A58.4200.1092   ;3238  ALU_R[0B],FLUSH.IB  ; LOAD THE VIBA WITH NON-EXISTANT ADDRESS
U 1092. 3000.003D.0180.6000.0000.11D8   ;3239 =0 LOAD.IB,IBC/START,CALL,J/WAITFORTIMEOUT ; GO WAIT FOR THE TIMEOUT
U 1093. 1000.003C.01C0.0A78.0000.1306   ;3240  IBC/STOP,Q_R[0F]  ; CHECK THE MICRO TRAP FLAG
U 1306. 0810.0038.0180.0970.0000.1307   ;3241  D_RC[0E]
U 1307. 001D.2000.0180.0800.0010.1308   ;3242  ALU_Q-D,CLK.UBCC  ; CHECK FOR NO MICRO TRAP
U 1308. 0000.013C.0180.0800.0000.1094   ;3243  Z?    ; BRANCH IF NO MICRO TRAP
```

```
            ;3244 =0
U 1094, 0001,203D,0180,09E8,0000,1149  ;3245 ERR7: ERROR2,RC[0D]_Q   ; TIMEOUT TRAP DID NOT OCCUR
            ;3246
            ;3247 ;*
            ;3248 ; NOW CHECK THAT THE INTERRUPT DID NOT OCCUR
            ;3249 ;-
            ;3250
U 1095, 0000,003C,0180,0800,4000,1309  ;3251  INTRPT.STROBE   ; LATCH IT IF ITS THERE
U 1309, 0000,003C,35F0,2C00,0000,130A  ;3252  Q_ID[VECTOR]    ; READ THE TIMEOUT VECTOR
U 130A, 0019,2034,49C0,0800,0010,130B  ;3253  Q_Q.AND.K[.FF],CLK.UBCC  ; MASK AND CHECK FOR ZERO
U 130B, 0003,013C,0180,09F0,0000,1096  ;3254  Z?,RC[0E]_0    ; BRANCH IF OK
            ;3255 =0
U 1096, 0001,203D,0180,09E8,0000,1149  ;3256 ERR8: ERROR2,RC[0D]_Q   ; INTERRUPT VECTOR INCORRECT
            ;3257
            ;3258 ;*
            ;3259 ; NOW CHECK THE TIMEOUT BIT IN THE ERROR REGISTER
            ;3260 ;-
            ;3261
U 1097, 0000,003C,65F0,2C00,0000,130C  ;3262  Q_ID[SBI.ERR]    ; READ THE TIMEOUT BIT
U 130C, 0818,0038,3180,0800,0000,130D  ;3263  D_K[.40]    ; GENERATE EXPECTED ERROR REG
U 130D, 0819,0030,0980,09F0,0000,130E  ;3264  RC[0E]_D.OR.K[.2],D_ALU  ; ...
U 130E, 001D,2000,0180,0800,0010,130F  ;3265  ALU_Q-D,CLK.UBCC  ; CHECK THE REGISTER
U 130F, 0000,013C,0180,0800,0000,1098  ;3266  Z?     ; BRANCH IF OK
            ;3267 =0
U 1098, 0001,203D,0180,09E8,0000,1149  ;3268 ERR9: ERROR2,RC[0D]_Q   ; SBI ERROR REGISTER INCORRECT
            ;3269
            ;3270 ;*
            ;3271 ; NOW CHECK THAT THE ERROR REGISTER CLEARS
            ;3272 ;-
            ;3273
U 1099, 0818,0038,3180,0800,0000,1310  ;3274  D_K[.40]    ; GENERATE PATTERN TO CLEAR IT
U 1310, 0000,003C,6580,3C00,0000,1311  ;3275  ID[SBI.ERR]_D    ; TRY AND CLEAR THE REGISTER
U 1311, 0000,003C,65F0,2C00,0000,1312  ;3276  Q_ID[SBI.ERR]    ; READ IT BACK
U 1312, 0818,0038,0980,09F0,0000,1313  ;3277  RC[0E]_K[.2],D_ALU  ; GENERATE EXPECTED RESULT
U 1313, 001D,2000,0180,0800,0010,1314  ;3278  ALU_Q-D,CLK.UBCC  ; CHECK THE REGISTER
U 1314, 0000,013C,0180,0800,0000,109A  ;3279  Z?     ; BRANCH IF OK
            ;3280 =0
U 109A, 0001,203D,0180,09E8,0000,1149  ;3281 ERRA: ERROR2,RC[0D]_Q   ; SBI ERROR REGISTER DID NOT CLEAR
            ;3282
            ;3283 ;*
            ;3284 ; NOW CHECK THAT THE IB DID NOT RECEIVE ANY DATA
            ;3285 ;-
            ;3286
U 109B, 0800,003C,0180,0A00,0000,109C  ;3287  D_R[0]     ; GET EXPECTED UPC
U 109C, 0001,003D,0180,09F0,0000,115E  ;3288 =0 RC[0E]_D,CALL,J/DISPAT  ; EXECUTE THE CALL3
U 109D, 001D,2000,0180,0800,0010,1315  ;3289  ALU_Q-D,CLK.UBCC  ; CHECK THE UPC
U 1315, 0000,013C,0180,0800,0000,109E  ;3290  Z?     ; BRANCH IF OK
            ;3291 =0
U 109E, 0001,203D,0180,09E8,0000,1149  ;3292 ERRB: ERROR2,RC[0D]_Q   ; UPC INCORRECT
U 109F, 0000,003C,0180,0800,0000,10A0  ;3293 END003: NOP
            ;3294
```

```
                ;3295  .PAGE "TEST 1DE INSTRUCTION BUFFER READ HIT AND ADDRESS NOT VALID"
                ;3296 ;********************************************************************************
                ;3297 ;*
                ;3298 ;
                ;3299 ; INSTRUCTION BUFFER READ HIT AND ADDRESS NOT VALID
                ;3300 ;
                ;3301 ; TEST DESCRIPTION
                ;3302 ;
                ;3303 ; THIS TEST CHECKS THAT "CANCEL" (GENERATED FROM ADDRESS NOT VALID)
                ;3304 ; INHIBITS THE DATA FROM BEING SENT TO THE IB WITH A CACHE HIT.
                ;3305 ; IT ALSO CHECKS THAT "CANCEL" INHIBITS THE LOADING OF THE HIT/MISS
                ;3306 ; BITS.
                ;3307 ;
                ;3308 ; LOGIC DESCRIPTION
                ;3309 ;
                ;3310 ; THIS TEST CHECKS THE LOGIC ON TBM THAT GENERATES "CANCEL"
                ;3311 ; DUE TO "ADDRESS NOT VALID" AND THE LOGIC ON SBL THAT INHIBITS
                ;3312 ; "IB READ DATA" AND "CLOCK HIT REGISTER" WHEN CANCEL IS ASSERTED.
                ;3313 ;
                ;3314 ; ERROR DESCRIPTION
                ;3315 ;
                ;3316 ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
                ;3317 ;
                ;3318 ; ERRC: DATA EXPECTED UPC WHEN "SUB/SPEC" ASCERTED
                ;3319 ;   RECEIVED UPC WHEN "SUB/SPEC" ASCERTED
                ;3320 ; ERRD: DATA EXPECTED HIT/MISS BITS
                ;3321 ;   RECEIVED HIT/MISS BITS
                ;3322 ;-
                ;3323 ;********************************************************************************
                ;3324
                ;3325 =0
U 10A0. 0000,003D,0180,0800,0000,1131  ;3326 T004: NEWTST
U 10A1. 0000,003C,0180,0800,0000,10A2  ;3327  NOP
U 10A2. 0000,003D,0180,0800,0000,102A  ;3328 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10A3. 0000,003C,0180,0800,0000,10A4  ;3329  NOP
U 10A4. 0818,0039,5D80,0800,0000,1154  ;3330 =0 D_K[.7],CALL,J/DCE  ; GENERATE EXPECTED UPC
U 10A5. 0001,003C,0180,0A80,0000,10A6  ;3331  R[0]_D    ; SAVE
U 10A6. 0000,003D,0180,0800,0000,1146  ;3332 =0 ERLOOP
U 10A7. 0F18,0038,1980,0800,0200,1316  ;3333  ALU_0(K),VA_ALU,D_0  ; LOAD THE VA
U 1316. 0000,003C,0180,9000,0000,1317  ;3334  MCT/INVALIDATE   ; SETUP TO INIT THE HIT/MISS BITS
U 1317. 0000,003C,0180,C800,0000,1318  ;3335  D[LONG]_CACHE.P   ; SETUP HIT/MISS BITS AND MAKE ADDRESS
                ;3336       ; ZERO A HIT
U 1318. 2018,0038,1980,0800,4200,1319  ;3337  ALU_0(K),FLUSH.IB  ; LOAD THE VIBA
U 1319. 0000,003C,0180,6000,0000,131A  ;3338  LOAD.IB    ; LOAD THE IPA
U 131A. 2000,003C,0180,0800,0000,131B  ;3339  IBC/FLUSH   ; ASSERT "ADDRESS NOT VALID"
U 131B. 3000,003C,0180,F800,0000,131C  ;3340  IBC/START,MCT/ALLOW.IB.READ ; START THE IB
U 131C. 0000,003C,0180,F800,0000,131D  ;3341  MCT/ALLOW.IB.READ  ; WAIT A CYCLE
U 131D. 1800,003C,0180,FA00,0000,10A8  ;3342  MCT/ALLOW.IB.READ,IBC/STOP,D_R[0] ; STOP THE IB
U 10A8. 0001,003D,0180,09F0,0000,115E  ;3343 =0 RC[0E]_D,CALL,J/DISPAT  ; GO EXECUTE THE CALL3
U 10A9. 001D,2000,0180,0800,0010,131E  ;3344  ALU_Q-D,CLK.UBCC  ; CHECK THE RECEIVED UPC
U 131E. 0000,013C,0180,0800,0000,10AA  ;3345  Z?    ; BRANCH IF OK
                ;3346 =0
U 10AA. 0001,203D,0180,09E8,0000,1149  ;3347 ERRC: ERROR2,RC[0D]_Q   ; UPC INCORRECT
                ;3348
                ;3349 ;*
```

```
            ;3350 ; CHECK THAT THE HIT/MISS BITS DIDN'T CHANGE
            ;3351 ;-
            ;3352
U 10AB. 0F00.003C.75F0.2C00.0000.131F  ;3353  Q_ID[MAINT],D_0   ; READ THE BITS
U 131F. 001C.0034.01C0.0A38.0010.1320  ;3354  Q_Q.AND.R[7],CLK.UBCC  ; MASK AND CHECK
U 1320. 0003.013C.0180.09F0.0000.10AC  ;3355  Z?,RC[0E]_0   ; BRANCH IF OK
            ;3356 =0
U 10AC. 0001.203D.0180.09E8.0000.1149  ;3357 ERRD: ERROR2,RC[0D]_Q   ; HIT/MISS BITS INCORRECT
U 10AD. 0000.003C.0180.0800.0000.10AE  ;3358 END004: NOP
            ;3359
```

```
                ;3360  .PAGE "TEST 1DF INSTRUCTION BUFFER READ MISS AND CACHE PARITY ERROR"
                ;3361 ;*************************************************************************
                ;3362 ;*
                ;3363 ;
                ;3364 ; INSTRUCTION BUFFER READ MISS AND CACHE PARITY ERROR
                ;3365 ;
                ;3366 ; TEST DESCRIPTION
                ;3367 ;
                ;3368 ; THIS TEST CHECKS THAT AN IB READ MISS WITH A CACHE PARITY ERROR
                ;3369 ; DOES NOT START AN SBI CYCLE AND THAT THE IB DOES NOT RECEIVE ANY
                ;3370 ; DATA.
                ;3371 ;
                ;3372 ; LOGIC DESCRIPTION
                ;3373 ;
                ;3374 ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT INHIBITS THE
                ;3375 ; ASSERTION OF "RAISE TR FF" WITH AN IB READ MISS AND CACHE PARITY
                ;3376 ; ERROR.
                ;3377 ;
                ;3378 ; ERROR DESCRIPTION
                ;3379 ;
                ;3380 ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
                ;3381 ;
                ;3382 ; ERRE: DATA EXPECTED VALUE OF THE "SBI INTERFACE NOT BUSY" BIT
                ;3383 ;   RECEIVED VALUE OF THE "SBI INTERFACE NOT BUSY" BIT
                ;3384 ; ERRF: DATA EXPECTED UPC WHEN "SUB/SPEC" ASCERTED
                ;3385 ;   RECEIVED UPC WHEN "SUB/SPEC" ASCERTED
                ;3386 ;-
                ;3387 ;*************************************************************************
                ;3388
                ;3389 =0
U 10AE. 0000.003D.0180.0800.0000.1131  ;3390 T005: NEWTST
U 10AF. 0000.003C.0180.0800.0000.10B0  ;3391  NOP
U 10B0. 0000.003D.0180.0800.0000.102A  ;3392 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10B1. 0000.003C.0180.0800.0000.10B2  ;3393  NOP
U 10B2. 0818.0039.5D80.0800.0000.1153  ;3394 =0 D_K[.7],CALL,J/DCD  ; GENERATE EXPECTED UPC
U 10B3. 0001.003C.0180.0A80.0000.10B4  ;3395  R[0]_D    ; SAVE
U 10B4. 0000.003D.0180.0800.0000.1146  ;3396 =0 ERLOOP
U 10B5. 0018.0038.1980.0800.0200.1321  ;3397  ALU_0(K),VA_ALU   ; LOAD THE VA
U 1321. 0818.0038.0580.0800.0000.1322  ;3398  D_K[.1]
U 1322. 0000.003C.0180.9000.0000.1323  ;3399  MCT/INVALIDATE   ; MAKE ADDRESS 0 A MISS
U 1323. 2018.0038.1980.0800.4200.1324  ;3400  ALU_0(K),FLUSH.IB  ; LOAD THE VIBA
U 1324. 0800.003C.0180.0A50.0000.1325  ;3401  D_R[0A]
U 1325. 0000.003C.7580.3C00.0000.1326  ;3402  ID[MAINT]_D   ; SET REVERSE PARITY BITS
U 1326. 3000.003C.0180.6000.0000.1327  ;3403  LOAD.IB,IBC/START  ; START THE IB
U 1327. 0000.003C.0180.F800.0000.1328  ;3404  MCT/ALLOW.IB.READ  ; WAIT A CYCLE
U 1328. 0000.003C.65F0.2C00.0000.10B6  ;3405  Q_ID[SBI.ERR]   ; READ THE BUSY BIT
U 10B6. 0001.203D.0180.FA88.0000.11D4  ;3406 =0 R[1]_Q,CALL,J/WAIT7,MCT/ALLOW.IB.READ ; WAIT SOME MORE
U 10B7. 1818.0038.0980.09F0.0000.1329  ;3407  RC[0E]_K[.2],D_ALU,IBC/STOP ; STOP THE IB
U 1329. 0018.0034.09C0.0A08.0000.132A  ;3408  Q_R[1].AND.K[.2]  ; CHECK THAT BUSY BIT DID NOT CLEAR
U 132A. 001D.2000.0180.0800.0010.132B  ;3409  ALU_Q-D,CLK.UBCC
U 132B. 0000.013C.0180.0800.0000.10B8  ;3410  Z?    ; BRANCH IF OK
                ;3411 =0
U 10B8. 0001.203D.0180.09E8.0000.1149  ;3412 ERRE: ERROR2,RC[0D]_Q   ; SBI CYCLE GOT STARTED
                ;3413
                ;3414 ;*
```

```
            ;3415 ; NOW CHECK THAT THE IB DIDN'T GET ANY DATA
            ;3416 ;-
            ;3417
U 10B9. 0800.003C.0180.0A00.0000.10BA  ;3418  D_R[0]     ; GET THE EXPECTED UPC
U 10BA. 0001.003D.0180.09F0.0000.115E  ;3419 =0 RC[0E]_D,CALL,J/DISPAT  ; GO EXECUTE THE CALL3
U 10BB. 001D.2000.0180.0800.0010.132C  ;3420  ALU_Q-D,CLK.UBCC  ; CHECK THE UPC
U 132C. 0000.013C.0180.0800.0000.10BC  ;3421  Z?     ; BRANCH IF OK
            ;3422 =0
U 10BC. 0001.203D.0180.09E8.0000.1149  ;3423 ERRF: ERROR2,RC[0D]_Q   ; UPC INCORRECT
U 10BD. 0000.003C.0180.0800.0000.10BE  ;3424 END005: NOP
            ;3425
            ;3426
```

```
                ;3427  .PAGE "TEST 1E0 IB READ MISS AND CP READ HIT"
                ;3428 ;*****************************************************************************
                ;3429 ;**
                ;3430 ;
                ;3431 ; IB READ MISS AND CP READ HIT
                ;3432 ;
                ;3433 ; TEST DESCRIPTION
                ;3434 ;
                ;3435 ; THIS TEST CHECKS THAT IF THE CPU EXECUTES A READ HIT WHILE THE
                ;3436 ; SBI IS BUSY EXECUTING AN IB READ MISS THAT NO STALL OCCURS. THIS IS
                ;3437 ; DETECTED BY INITIALIZING THE SILO COUNTER TO ZERO, STARTING THE CP
                ;3438 ; READ, THEN READING THE SILO COUNTER AND CHECKING FOR THE CORRECT
                ;3439 ; NUMBER OF CLOCK TICKS.
                ;3440 ;
                ;3441 ; LOGIC DESCRIPTION
                ;3442 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
                ;3443 ;
                ;3444 ; ERROR DESCRIPTION
                ;3445 ;
                ;3446 ; DATA: EXPECTED SILO COUNTER
                ;3447 ;       RECEIVED SILO COUNTER
                ;3448 ;--
                ;3449 ;*****************************************************************************
                ;3450 =0
U 10BE. 0000.003D.0180.0800.0000.1131  ;3451 T006: NEWTST
U 10BF. 0000.003C.0180.0800.0000.10C0  ;3452  NOP
U 10C0. 0000.003D.0180.0800.0000.102A  ;3453 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10C1. 0818.0038.0980.0800.0000.132D  ;3454  D_K[.2]    ; GENERATE EXPECTED DATA
U 132D. 0000.003C.65F8.0800.0084.732E  ;3455  SC_K[.10],Q_0   ; ...
U 132E. 0D00.003C.0180.0800.0000.132F  ;3456  D_DAL.SC   ; ...
U 132F. 0001.003C.0180.09F0.0000.1330  ;3457  RC[0E]_D   ;
U 1330. 0F18.0038.1980.0800.0200.1331  ;3458 RETRY2: VA_K[ZERO],D_0   ; LOAD THE VA
U 1331. 0000.003C.0180.A800.0000.1332  ;3459  CACHE.P_D[LONG]   ; WRITE THE DATA PATTERN TO MEMORY
U 1332. 0000.003C.0180.0803.0000.1333  ;3460  VA_VA+4
U 1333. 0000.003C.0180.A800.0000.1334  ;3461  CACHE.P_D[LONG]   ; ...
U 1334. 0018.0038.1980.0800.0200.1335  ;3462  VA_K[ZERO]   ; SETUP TO MAKE ADR 0 AND 4 A HIT
U 1335. 0000.003C.0180.C800.0000.10C2  ;3463  D[LONG]_CACHE.P   ; MAKE ADR 0 AND 4 A HIT
U 10C2. 0000.003D.0180.0800.0000.1146  ;3464 =0 ERLOOP
U 10C3. 0018.0038.0180.0800.0200.1336  ;3465  VA_K[.8]   ; SETUP TO MAKE ADR 8 AND C A MISS
U 1336. 0000.003C.0180.9000.0000.1337  ;3466  CACHE.INVALIDATE  ; INVALIDATE ADR 8
U 1337. 0000.003C.0180.0803.0000.1338  ;3467  VA_VA+4
U 1338. 0000.003C.0180.9000.0000.10C4  ;3468  CACHE.INVALIDATE  ; AND ADR C
U 10C4. 0000.003D.0180.0800.0000.11DC  ;3469 =0 CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 10C5. 2018.0038.0180.0800.4200.1339  ;3470  ALU_K[.8],FLUSH.IB  ; LOAD THE VIBA
U 1339. 3818.0038.7580.6000.0000.133A  ;3471  LOAD.IB,IBC/START,D_K[.20] ; LOAD THE IBA
U 133A. 0B18.0038.1980.F800.0200.133B  ;3472  D_D.SWAP,ALU_K[ZERO],VA_ALU,MCT/ALLOW.IB.READ ; LOAD THE VA FOR THE CP READ
U 133B. 0000.003C.7180.3C00.0000.133C  ;3473  ID[COMP]_D   ; INIT THE SILO COUNTER
U 133C. 0000.003C.0180.F800.0000.133D  ;3474  MCT/ALLOW.IB.READ  ; LET THE IB REFERANCE START THE SBI
U 133D. 0000.003C.0180.C800.0000.133E  ;3475  D[LONG]_CACHE.P   ; EXECUTE THE CP READ HIT
U 133E. 1000.003C.71F0.2C00.0000.133F  ;3476  Q_ID[COMP],IBC/STOP  ; READ THE SILO COUNTER
U 133F. C01C.0034.01C0.0A60.0000.1340  ;3477  Q_Q.AND.R[0C]   ; MASK
U 1340. 0810.0038.0180.0970.0000.1341  ;3478  D_RC[0E]   ; GET EXPECTED VALUE
U 1341. 001D.2000.0180.0800.0010.1342  ;3479  ALU_Q-D,CLK.UBCC  ; CHECK THE SILO COUNTER
U 1342. 0001.213C.0180.09E8.0000.10C6  ;3480  Z?,RC[0D]_Q   ; BRANCH IF OK
U 10C6. 0000.003D.0180.0800.0000.1149  ;3481 =0 ERROR2    ; FAILURE AND NO REFRESH
```

```
U 10C7, 0000,003C,0180,0800,0000,10C8  ;3482 END1: NOP
            ;3483
            ;3484
```

```
            ;3485  .PAGE "TEST 1E1 INSTRUCTION BUFFER READ MISS AND CP READ MISS"
            ;3486 ;*****************************************************************************
            ;3487 ;**
            ;3488 ;
            ;3489 ; INSTRUCTION BUFFER READ MISS AND CP READ MISS
            ;3490 ;
            ;3491 ; TEST DESCRIPTION
            ;3492 ;
            ;3493 ; THIS TEST CHECKS THAT IF A CP READ MISS IS EXECUTED DURING AN IB
            ;3494 ; READ MISS THAT THE SBI STALLS UNTIL THE IB REFERANCE IS FINISHED.
            ;3495 ; THIS IS DONE BY STARTING THE IB, INITIALIZING THE SILO COUNTER, STARTING
            ;3496 ; THE CP READ, AND READING THE SILO COUNTER TO CHECK THAT APPROXIMATELY
            ;3497 ; TWO MEMORY REFERANCES WERE PEROFRMED.
            ;3498 ;
            ;3499 ; LOGIC DESCRIPTION
            ;3500 ;
            ;3501 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
            ;3502 ;
            ;3503 ; ERROR DESCRIPTION
            ;3504 ;
            ;3505 ; DATA: EXPECTED SILO COUNTER
            ;3506 ;       RECEIVED SILO COUNTER
            ;3507 ;--
            ;3508 ;*****************************************************************************
            ;3509 =0
U 10C8, 0000,003D,0180,0800,0000,1131  ;3510 T007: NEWTST
U 10C9, 0000,003C,0180,0800,0000,10CA  ;3511  NOP
U 10CA, 0000,003D,0180,0800,0000,102A  ;3512 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10CB, 0818,0038,85F8,0800,0000,1343  ;3513  D_K[.C],Q_0   ; GENERATE EXPECTED SILO COUNT
U 1343, 0000,003C,6580,0800,0084,7344  ;3514  SC_K[.10]
U 1344, 0D00,003C,0180,0800,0000,1345  ;3515  D_DAL.SC
U 1345, 0001,003C,0180,09F0,0000,1346  ;3516  RC[0E]_D   ;
U 1346, 0F18,0038,1980,0800,0200,1347  ;3517 RETRY3: ALU_0(K),VA_ALU,D_0  ; LOAD THE VA
U 1347, 0000,003C,0180,9000,0000,1348  ;3518  CACHE.INVALIDATE
U 1348, 0000,003C,0180,0803,0000,1349  ;3519  VA_VA+4
U 1349, 0000,003C,0180,9000,0000,134A  ;3520  CACHE.INVALIDATE
U 134A, 0000,003C,0180,0803,0000,134B  ;3521  VA_VA+4
U 134B, 0000,003C,0180,9000,0000,134C  ;3522  CACHE.INVALIDATE
U 134C, 0000,003C,0180,0803,0000,134D  ;3523  VA_VA+4
U 134D, 0000,003C,0180,9000,0000,10CC  ;3524  CACHE.INVALIDATE
U 10CC, 0000,003D,0180,0800,0000,11DC  ;3525 =0 CALL[WAITREFRESH]
U 10CD, 2018,0038,0180,0800,4200,134E  ;3526  ALU_K[.8],FLUSH.IB  ; LAOD THE VIBA
U 134E, 3818,0038,7580,6000,0000,134F  ;3527  LOAD.IB,IBC/START,D_K[.20] ; START THE IB
U 134F, 0B18,0038,1980,F800,0200,1350  ;3528  ALU_0(K),VA_ALU,D_D.SWAP,MCT/ALLOW.IB.READ ; LOAD THE VA
U 1350, 0900,003C,7180,3C00,0000,1351  ;3529  ID[COMP]_D   ; INIT THE SILO COUNTER
U 1351, 0000,003C,0180,C800,0000,1352  ;3530  D[LONG]_CACHE.P   ; EXECUTE THE READ MISS
U 1352, 1000,003C,71F0,2C00,0000,1353  ;3531  Q_ID[COMP],IBC/STOP  ; READ THE SILO COUNTER
U 1353, 001C,0034,01C0,0A60,0000,1354  ;3532  Q_Q.AND.R[0C]   ; MASK
U 1354, 0810,003B,0180,0970,0000,1355  ;3533  D_RC[0E]   ; GET THE EXPECTED DATA
U 1355, 001D,2000,0180,0800,0010,1356  ;3534  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED DATA
U 1356, 0000,013C,0180,0800,0000,10CE  ;3535  Z?    ; BRANCH IF OK
U 10CE, 0000,003D,0180,0800,0000,1149  ;3536 =0 ERROR2    ; FAILURE NOT DUE TO REFRESH
U 10CF, 0000,003C,0180,0800,0000,10D0  ;3537 END2: NOP
            ;3538
            ;3539
```

```
            ;3540  .PAGE "TEST 1E2 INSTRUCTION BUFFER READ MISS AND CP WRITE"
            ;3541 ;*****************************************************************************
            ;3542 ;**
            ;3543 ;
            ;3544 ; INSTRUCTION READ MISS AND CP WRITE
            ;3545 ;
            ;3546 ; TEST DESCRIPTION
            ;3547 ;
            ;3548 ; THIS TEST CHECKS THAT A CP WRITE IS STALLED UNTIL THE INSTRUCTION
            ;3549 ; BUFFER READ IS COMPLETE. THIS IS DONE BY STARTING THE IB, INITIALIZING
            ;3550 ; THE SILO COUNTER, STARTINE THE CP WRITE, THEN READING
            ;3551 ; AND CHECKING THE SILO COUNTER FOR THE CORRECT NUMBER OF CYCLES.
            ;3552 ;
            ;3553 ; LOGIC DESCRIPTION
            ;3554 ;
            ;3555 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE
            ;3556 ;
            ;3557 ; ERROR DESCRIPTION
            ;3558 ;
            ;3559 ; DATA: EXPECTED SILO COUNTER
            ;3560 ;       RECEIVED SILO COUNTER
            ;3561 ;--
            ;3562 ;*****************************************************************************
            ;3563 =0
U 10D0, 0000,003D,0180,0800,0000,1131   ;3564 T008: NEWTST
U 10D1, 0000,003C,0180,0800,0000,10D2   ;3565  NOP
U 10D2, 0000,003D,0180,0800,0000,102A   ;3566 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10D3, 0818,0038,D5F8,0800,0000,1357   ;3567  D_K[.6],Q_0
U 1357, 0000,003C,6580,0800,0084,7358   ;3568  SC_K[.10]
U 1358, 0D00,003C,0180,0800,0000,1359   ;3569  D_DAL.SC
U 1359, 0001,003C,0180,09F0,0000,135A   ;3570  RC[0E]_D   ; GENERATE EXPECTED DATA
U 135A, 0F18,0038,0180,0800,0200,135B   ;3571 RETRY4: ALU_K[.8],VA_ALU,D_0  ; LOAD THE VA
U 135B, 0000,003C,0180,9000,0000,135C   ;3572  CACHE.INVALIDATE
U 135C, 0000,003C,0180,0803,0000,135D   ;3573  VA_VA+4
U 135D, 0000,003C,0180,9000,0000,10D4   ;3574  CACHE.INVALIDATE
U 10D4, 0000,003D,0180,0800,0000,11DC   ;3575 =0 CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 10D5, 2018,0038,0180,0800,4200,135E   ;3576  ALU_K[.8],FLUSH.IB  ; LAOD THE VIBA
U 135E, 3818,0038,7580,6000,0000,135F   ;3577  LOAD.IB,IBC/START,D_K[.20] ; START THE IB
U 135F, 0B18,0038,1980,F800,0200,1360   ;3578  ALU_0(K),VA_ALU,D_D.SWAP,MCT/ALLOW.IB.READ ; LOAD THE VA
U 1360, 0000,003C,7180,3C00,0000,1361   ;3579  ID[COMP]_D   ; INIT THE SILO COUNTER
U 1361, 0000,003C,0180,A800,0000,1362   ;3580  CACHE.P_D[LONG]   ; EXECUTE THE WRITE
U 1362, 1000,003C,71F0,2C00,0000,1363   ;3581  Q_ID[COMP],IBC/STOP  ; READ THE SILO COUNTER
U 1363, 001C,0034,01C0,0A60,0000,1364   ;3582  Q_Q.AND.R[0C]   ; MASK
U 1364, 0810,0038,0180,0970,0000,1365   ;3583  D_RC[0E]   ; GET THE EXPECTED DATA
U 1365, 001D,2000,0180,0800,0010,1366   ;3584  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED DATA
U 1366, 0001,213C,0180,09E8,0000,10D6   ;3585  Z?,RC[0D]_Q   ; BRANCH IF OK
U 10D6, 0000,003D,0180,0800,0000,1149   ;3586 =0 ERROR2    ; FAILURE NOT DUE TO REFRESH
U 10D7, 0000,003C,0180,0800,0000,10D8   ;3587 END3: NOP
            ;3588
            ;3589
```

```
                ;3590  .PAGE "TEST 1E3 INSTRUCTION BUFFER MISS AND CP VALIDATE"
                ;3591 ;*****************************************************************
                ;3592 ;**
                ;3593 ;
                ;3594 ; INSTRUCTION BUFFER MISS AND CP VALIDATE
                ;3595 ;
                ;3596 ; TEST DESCRIPTION
                ;3597 ;
                ;3598 ; THIS TEST CHECKS THAT A CP VALIDATE IS STALLED UNTIL THE INSTRUCTION
                ;3599 ; BUFFER READ IS COMPLETE. THIS IS DONE BY STARTING THE IB, INITIALIZING
                ;3600 ; THE SILO COUNTER, STARTING THE CP VALIDATE, THEN READING AND CHECKING
                ;3601 ; THE SILO COUNTER FOR THE CORRECT NUMBER OF CYCLES.
                ;3602 ;
                ;3603 ; LOGIC DESCRIPTION
                ;3604 ;
                ;3605 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
                ;3606 ;
                ;3607 ; ERROR DESCRIPTION
                ;3608 ;
                ;3609 ; DATA: EXPECTED SILO COUNTER
                ;3610 ;       RECEIVED SILO COUNTER
                ;3611 ;--
                ;3612 ;*****************************************************************
                ;3613 =0
U 10D8. 0000.003D.0180.0800.0000.1131  ;3614 T008A: NEWTST
U 10D9. 0000.003C.0180.0800.0000.10DA  ;3615  NOP
U 10DA. 0000.003D.0180.0800.0000.102A  ;3616 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10DB. 0818.0038.D5F8.0800.0000.1367  ;3617  D_K[.6],Q_0
U 1367. 0000.003C.6580.0800.0084.7368  ;3618  SC_K[.10]
U 1368. 0D00.003C.0180.0800.0000.1369  ;3619  D_DAL.SC
U 1369. 0001.003C.0180.09F0.0000.136A  ;3620  RC[0E]_D   ; GENERATE EXPECTED DATA
U 136A. 0F18.0038.0180.0800.0200.136B  ;3621 RETRY5: ALU_K[.8],VA_ALU,D_0  ; LOAD THE VA
U 136B. 0000.003C.0180.9000.0000.136C  ;3622  CACHE.INVALIDATE
U 136C. 0000.003C.0180.0803.0000.136D  ;3623  VA_VA+4
U 136D. 0000.003C.0180.9000.0000.10DC  ;3624  CACHE.INVALIDATE
U 10DC. 0000.003D.0180.0800.0000.11DC  ;3625 =0 CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 10DD. 2018.0038.0180.0800.4200.136E  ;3626  ALU_K[.8],FLUSH.IB  ; LAOD THE VIBA
U 136E. 3818.0038.7580.6000.0000.136F  ;3627  LOAD.IB,IBC/START,D_K[.20] ; START THE IB
U 136F. 0B18.0038.1980.F800.0200.1370  ;3628  ALU_0(K),VA_ALU,D_D.SWAP,MCT/ALLOW.IB.READ ; LOAD THE VA
U 1370. 0000.003C.7180.3C00.0000.1371  ;3629  ID[COMP]_D   ; INIT THE SILO COUNTER
U 1371. 0000.003C.0180.9800.0000.1372  ;3630  MCT/VALIDATE   ; EXECUTE THE VALIDATE
U 1372. 1000.003C.71F0.2C00.0000.1373  ;3631  Q_ID[COMP],IBC/STOP  ; READ THE SILO COUNTER
U 1373. 001C.0034.01C0.0A60.0000.1374  ;3632  Q_Q.AND.R[0C]   ; MASK
U 1374. 0810.0038.0180.0970.0000.1375  ;3633  D_RC[0E]   ; GET THE EXPECTED DATA
U 1375. 001D.2000.0180.0800.0010.1376  ;3634  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED DATA
U 1376. 0001.213C.0180.09E8.0000.10DE  ;3635  Z?,RC[0D]_Q   ; BRANCH IF OK
U 10DE. 0000.003D.0180.0800.0000.1149  ;3636 =0 ERROR2    ; FAILURE NOT DUE TO REFRESH
U 10DF. 0000.003C.0180.0800.0000.10E0  ;3637 END4: NOP
                ;3638
                ;3639
```

```
                ;3640  .PAGE "TEST 1E4 INSTRUCTION BUFFER MISS AND RDS"
                ;3641  ;*****************************************************************************
                ;3642  ;+
                ;3643  ;
                ;3644  ; INSTRUCTION BUFFER MISS AND RDS
                ;3645  ;
                ;3646  ; TEST DESCRIPTION
                ;3647  ;
                ;3648  ; THIS TEST EXECUTES AN INSTRUCTION BUFFER READ MISS WHILE FORCING AN
                ;3649  ; RDS ERROR. IT CHECKS THAT THE RDS BIT SETS AND CLEARS IN THE ERROR
                ;3650  ; REGISTER, THAT NO MICRO TRAP OCCURS, THAT AN INTERRUPT OCCURS, AND
                ;3651  ; THAT THE IB DOES NOT GET ANY VALID DATA.
                ;3652  ;
                ;3653  ; LOGIC DESCRIPTION
                ;3654  ;
                ;3655  ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT GENERATES THE UTRAP,
                ;3656  ; INTERRUPT, ERROR REGISTER, AND IB READ DATA DURING AN IB READ
                ;3657  ; WITH AN RDS ERROR.
                ;3658  ;
                ;3659  ; ERROR DESCRIPTION
                ;3660  ;
                ;3661  ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
                ;3662  ;
                ;3663  ; ERR10: DATA EXPECTED MICRO TRAP FLAG
                ;3664  ;   RECEIVED MICRO TRAP FLAG
                ;3665  ; ERR11: DATA EXPECTED VECTOR REGISTER
                ;3666  ;   RECEIVED VECTOR REGISTER
                ;3667  ; ERR12: DATA EXPECTED SBI ERROR REGISTER
                ;3668  ;   RECEIVED SBI ERROR REGISTER
                ;3669  ; ERR13: DATA EXPECTED SBI ERROR REGISTER
                ;3670  ;   RECEIVED SBI ERROR REGISTER
                ;3671  ; ERR14: DATA EXPECTED UPC WHEN "SUB/SPEC" ACTIVE
                ;3672  ;   RECEIVED UPC WHEN "SUB/SPEC" ACTIVE
                ;3673  ;--
                ;3674  ;*****************************************************************************
                ;3675  =0
U 10E0, 0000.003D.0180.0800.0000.1131   ;3676 T009: NEWTST
U 10E1, 0000.003C.0180.0800.0000.10E2   ;3677  NOP
U 10E2, 0000.003D.0180.0800.0000.102A   ;3678 =0 CALL.J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10E3, 0000.003C.0180.0800.0000.10E4   ;3679  NOP
U 10E4, 0818.0039.0D80.0800.0000.1152   ;3680 =0 D_K[.3],CALL.J/DCC  ; GENERATE EXPECTED UPC
U 10E5, 0001.003C.0180.0A88.0000.1377   ;3681  R[1]_D    ; SAVE
U 1377, 0F00.003C.0180.0800.0000.1378   ;3682  D_0
U 1378, 0000.003C.3D80.3C00.0000.10E6   ;3683  ID[PSL]_D
U 10E6, 0000.003D.0180.0800.0000.1146   ;3684 =0 ERLOOP
U 10E7, 0018.0038.1980.0800.0200.1379   ;3685  VA_K[ZERO]   ; Point VA to locatio 0
U 1379, 0000.003C.0180.9000.0000.137A   ;3686  CACHE.INVALIDATE  ; INVALIDATE THE CACHE
U 137A, 0000.003C.0180.0A70.0200.137B   ;3687  VA_ALU,ALU_R[0E]  ; Point VA to CNFG B Reg
U 137B, 0800.003C.0180.0A68.0000.137C   ;3688  D_R[0D]    ; Get bits to force RDS
U 137C, 0000.003C.0180.A800.0000.137D   ;3689  CACHE.P_D[LONG]   ; Set force RDS bits in CNFG B
U 137D, 0818.0038.0D80.0800.0000.137E   ;3690  D_K[.3]    ; Get data to initialize location 0
U 137E, 0018.0038.1980.0800.0200.137F   ;3691  VA_K[ZERO]   ; Point VA to location 0
U 137F, 0000.003C.0180.A800.0000.1380   ;3692  CACHE.P_D[LONG]   ; Initialize location 0
                ;3693      ; ...with data for RDS
U 1380, 0000.003C.8580.0800.0084.7381   ;3694  SC_K[.C]   ; Enable RDS uTRAPS
```

```
U 1381. 0003.001C.0180.0AF8.0000.10E8  ;3695  R[0F]_NOT.MASK   ; SETUP THE EXPECTED TRAP FLAG
U 10E8. 0003.001D.0180.09F0.0000.11DC  ;3696 =0 CALL[WAITREFRESH],RC[0E]_NOT.MASK ; WAIT FOR REFRESH TO OCCUR
U 10E9. 0800.003C.0180.0A68.0000.1382  ;3697  D_R[0D]    ; Get force RDS bits for CNFG B. These
U 1382. 0000.003C.0180.A800.0000.1383  ;3698  CACHE.P_D[LONG]   ; need to be restored as the routine
          ;3699        ; WAITREFRESH clears these bits out
U 1383. 2018.0038.1980.0800.4200.1384  ;3700  ALU_K[ZERO],FLUSH.IB   ; LOAD THE VIBA
U 1384. 3000.003C.0180.6000.0000.1385  ;3701  LOAD.IB,IBC/START   ; START THE IB REFERANCE
U 1385. 0000.003C.0180.F800.0000.1386  ;3702  MCT/ALLOW.IB.READ   ; WAIT A CYCLE
U 1386. 0000.003C.0180.F800.0000.1387  ;3703  MCT/ALLOW.IB.READ
U 1387. 0000.003C.0180.F800.0000.10EA  ;3704  MCT/ALLOW.IB.READ   ; GET THE EXPECTED TRAP FLAG
U 10EA. 0000.003D.0180.F800.0000.11D4  ;3705 =0 CALL,J/WAIT7,MCT/ALLOW.IB.READ ; GO WAIT
U 10EB. 0F00.003C.0180.0A70.0200.1388  ;3706  ALU_R[0E],VA_ALU,D_0   ; SETUP TO CLEAR FORCE RDS
U 1388. 0000.003C.0180.A800.0000.1389  ;3707  CACHE.P_D[LONG]   ; CLEAR THE FORCE RDS
U 1389. 0000.003C.01C0.0A78.0000.138A  ;3708  Q_R[0F]    ; GET THE TRAP FLAG
U 138A. 0810.0038.0180.0970.0000.138B  ;3709  D_RC[0E]   ; GET EXPECTED TRAP FLAG
U 138B. 001D.2000.0180.0800.0010.138C  ;3710  ALU_Q-D,CLK.UBCC   ; CHECK THAT NO TRAP OCCURRED
U 138C. 0000.013C.0180.0800.0000.10EC  ;3711  Z?    ; BRANCH IF OK
          ;3712 =0
U 10EC. 0001.203D.0180.09E8.0000.1149  ;3713 ERR10: ERROR2,RC[0D]_Q   ; UTRAP OCCURRED
          ;3714
          ;3715 ;*
          ;3716 ; NOW CHECK THAT THE INTERRUPT IS ACTIVE
          ;3717 ;-
          ;3718
U 10ED. 0818.0038.41F8.0800.0000.138D  ;3719  D_K[.80],Q_0   ; GENERATE PATTERN TO SET ENABLE BIT
U 138D. 0000.003C.0180.0800.0084.738E  ;3720  SC_K[.8]   ; ...
U 138E. 0D00.003C.0180.0800.0000.138F  ;3721  D_DAL.SC   ; ...
U 138F. 0000.003C.6580.3C00.0000.1390  ;3722  ID[SBI.ERR]_D   ; ENABLE THE INTERRUPT
U 1390. 0818.0038.3580.0800.4000.1391  ;3723  D_K[.50],INTRPT.STROBE   ; GENERATE EXPECTED VECTOR
U 1391. 0819.0030.1180.09F0.0000.1392  ;3724  D_D.OR.K[.4],RC[0E]_ALU   ; ...
U 1392. 0000.003C.35F0.2C00.0000.1393  ;3725  Q_ID[VECTOR]   ; READ THE VECTOR
U 1393. 0019.2034.C1C0.0800.0000.1394  ;3726  Q_Q.AND.K[.FFFF]   ; MASK
U 1394. 001D.2000.0180.0800.0010.1395  ;3727  ALU_Q-D,CLK.UBCC   ; CHECK THE VECTOR
U 1395. 000D.013C.0180.0800.0000.10EE  ;3728  Z?    ; BRANCH IF OK
          ;3729 =0
U 10EE. 0001.203D.0180.09E8.0000.1149  ;3730 ERR11: ERROR2,RC[0D]_Q   ; RDS INTERRUPT DID NOT OCCUR
U 10EF. 0F00.003C.0180.0800.0000.1396  ;3731  D_0    ; CLEAR THE INTERRUPT ENABLE BIT
U 1396. 0000.003C.6580.3C00.0000.1397  ;3732  ID[SBI.ERR]_D   ; ...
          ;3733
          ;3734 ;*
          ;3735 ; NOW CHECK THE ERROR REGISTER
          ;3736 ;-
          ;3737
U 1397. 0818.0038.75F8.0800.0000.1398  ;3738  D_K[.20],Q_0   ; GENERATE EXPECTED VALUE
U 1398. 0000.003C.0180.0800.0084.7399  ;3739  SC_K[.8]
U 1399. 0D00.003C.0180.0800.0000.139A  ;3740  D_DAL.SC
U 139A. 0819.0030.4180.0800.0000.139B  ;3741  D_D.OR.K[.80]
U 139B. 0819.0030.0980.09F0.0000.139C  ;3742  D_D.OR.K[.2],RC[0E]_ALU   ; SAVE
U 139C. 0000.003C.65F0.2C00.0000.139D  ;3743  Q_ID[SBI.ERR]   ; READ THE ERROR REGISTER
U 139D. 001D.2000.0180.0800.0010.139E  ;3744  ALU_Q-D,CLK.UBCC   ; CHECK IT
U 139E. 0000.013C.0180.0800.0000.10F0  ;3745  Z?    ; BRANCH IF OK
          ;3746 =0
U 10F0. 0001.203D.0180.09E8.0000.1149  ;3747 ERR12: ERROR2,RC[0D]_Q   ; RDS BIT DID NOT SET IN ERROR REG
          ;3748
          ;3749 ;*
```

```
            ;3750 ; NOW CHECK THAT THE ERROR REGISTER CLEARS
            ;3751 ;-
            ;3752
U 10F1. 0818.0038.7580.0800.0000.139F  ;3753  D_K[.20]    ; GENERATE VALUE TO CLEAR ERR REG
U 139F. 0000.003C.01F8.0800.0084.73A0  ;3754  SC_K[.8],Q_0    ; ...
U 13A0. 0D00.003C.0180.0800.0000.13A1  ;3755  D_DAL.SC    ; EQUALS 2000(X)
U 13A1. 0000.003C.6580.3C00.0000.13A2  ;3756  ID[SBI.ERR]_D    ; CLEAR THE REGISTER
U 13A2. 0818.0038.0980.09F0.0000.13A3  ;3757  D_K[.2],RC[0E]_ALU  ; GET EXPECTED VALUE OF THE REGISTER
U 13A3. 0000.003C.65F0.2C00.0000.13A4  ;3758  Q_ID[SBI.ERR]    ; READ THE REGISTER
U 13A4. 001D.2000.0180.0800.0010.13A5  ;3759  ALU_Q-D,CLK.UBCC  ; CHECK
U 13A5. 0000.013C.0180.0800.0000.10F2  ;3760  Z?     ; BRANCH IF OK
            ;3761 =0
U 10F2. 0001.203D.0180.09E8.0000.1149  ;3762 ERR13: ERROR2,RC[0D]_Q    ; RDS BIT DID NOT CLEAR IN ERR REG
            ;3763
            ;3764 ;*
            ;3765 ; NOW CHECK THAT THE IB DID NOT GET ANY DATA
            ;3766 ;-
            ;3767
U 10F3. 0800.003C.0180.0A08.0000.10F4  ;3768  D_R[1]     ; GET THE EXPECTED UPC
U 10F4. 0001.003D.0180.09F0.0000.115E  ;3769 =0 RC[0E]_D,CALL,J/DISPAT  ; GO EXECUTE THE SUB/SPEC
U 10F5. 001D.2000.0180.0800.0010.13A6  ;3770  ALU_Q-D,CLK.UBCC  ; CHECK
U 13A6. 0000.013C.0180.0800.0000.10F6  ;3771  Z?     ; BRANCH IF OK
            ;3772 =0
U 10F6. 0001.203D.0180.09E8.0000.1149  ;3773 ERR14: ERROR2,RC[0D]_Q    ; IB RECEIVED VALID DATA
U 10F7. 0000.003C.0180.0800.0000.10F8  ;3774 END009: NOP
            ;3775
            ;3776
```

```
            ;3777  .PAGE "TEST 1E5 SBI ERROR REGISTER CLEAR WITH IB FLUSH"
            ;3778 ;*****************************************************************************
            ;3779 ;*
            ;3780 ;
            ;3781 ; SBI ERROR REGISTER CLEAR WITH FLUSH
            ;3782 ;
            ;3783 ; TEST DESCRIPTION
            ;3784 ;
            ;3785 ; THIS TEST CHECKS THAT AN IB FLUSH CLEARS THE IB ERROR BITS IN THE
            ;3786 ; SBI ERROR REGISTER. THIS IS DONE BY EXECUTING AN IB READ THAT TIMES
            ;3787 ; OUT AND A READ THAT CAUSES AN RDS, THEN EXECUTING A FLUSH AND
            ;3788 ; CHECKING THAT THE ERROR REGISTER CLEARED.
            ;3789 ;
            ;3790 ; SUBTEST 1 - CHECK THAT THE RDS BIT CLEARS
            ;3791 ; SUBTEST 2 - CHECK THAT THE TIMEOUT BIT CLEARS
            ;3792 ;
            ;3793 ; LOGIC DESCRIPTION
            ;3794 ;
            ;3795 ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT CLEARS THE SBI
            ;3796 ; ERROR REGISTER WITH AN IB FLUSH.
            ;3797 ;
            ;3798 ; ERROR DESCRIPTION
            ;3799 ;
            ;3800 ; DATA: EXPECTED ERROR REGISTER
            ;3801 ;       RECEIVED ERROR REGISTER
            ;3802 ;-
            ;3803 ;*****************************************************************************
            ;3804 =0
U 10F8, 000D,003D,0180,0800,0000,1131  ;3805 T00A: NEWTST
U 10F9, 0000,003C,0180,0800,0000,10FA  ;3806  NOP
U 10FA, 0000,003D,0180,0800,0000,102A  ;3807 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10FB, 0818,0038,75F8,0800,0000,13A7  ;3808  D_K[.20],Q_0   ; GENERATE THE EXPECTED DATA
U 13A7, 0000,003C,0180,0800,0084,73A8  ;3809  SC_K[.8]
U 13A8, 0D00,003C,0180,0800,0000,13A9  ;3810  D_DAL,SC
U 13A9, 0019,0030,0980,09F0,0000,10FC  ;3811  RC[0E]_D.OR.K[.2]  ; SAVE
U 10FC, 0000,003D,0180,0800,0000,115C  ;3812 =0 SUBTEST
U 10FD, 0000,003C,0180,0800,0000,10FE  ;3813  NOP
U 10FE, 0000,003D,0180,0800,0000,1146  ;3814 =0 ERLOOP
U 10FF, 0018,0038,1980,0800,0200,13AA  ;3815 RETRY7: ALU_K[ZERO],VA_ALU  ; LOAD ADRS OF RDS LOCATION
U 13AA, 0000,003C,0180,9000,0000,13AB  ;3816  CACHE.INVALIDATE  ; MAKE SURE ITS A MISS
U 13AB, 0000,003C,0180,0A70,0200,13AC  ;3817  VA_ALU,ALU_R[0E]  ; Point VA to CNFG Reg B
U 13AC, 0800,003C,0180,0A68,0000,13AD  ;3818  D_R[0D]    ; Get bits for CNFG Reg B to
            ;3819          ; ...force RDS error
U 13AD, 0000,003C,0180,A800,0000,13AE  ;3820  CACHE.P_D[LONG]   ; Load the register
U 13AE, 0018,0038,1980,0800,0200,13AF  ;3821  VA_K[ZERO]   ; Load the VA with location 0
U 13AF, 0818,0038,0D80,0800,0000,13B0  ;3822  D_K[.3]    ; Load the D register with init data
U 13B0, 0000,003C,0180,A800,0000,110A  ;3823  CACHE.P_D[LONG]   ; Init the memory location
U 110A, 0000,003D,0180,0800,0000,11DC  ;3824 =0 CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 110B, 0800,003C,0180,0A68,0000,13B1  ;3825  D_R[0D]    ; Get force RDS bits for CNFG B. These
U 13B1, 0000,003C,0180,A800,0000,13B2  ;3826  CACHE.P_D[LONG]   ; need to be restored as the routine
            ;3827          ; WAITREFRESH clears these bits out
U 13B2, 2018,0038,1980,0800,4200,13B3  ;3828  ALU_K[ZERO],FLUSH.IB  ; LOAD THE VIBA
U 13B3, 3000,003C,0180,6000,0000,13B4  ;3829  LOAD.IB,IBC/START  ; START THE IB REFERANCE
U 13B4, 0000,003C,0180,F800,0000,13B5  ;3830  MCT/ALLOW.IB.READ
U 13B5, 0000,003C,0180,F800,0000,13B6  ;3831  MCT/ALLOW.IB.READ
```

```
U 13B6. 0000.003C.0180.F800.0000.1110  ;3832  MCT/ALLOW.IB.READ
U 1110. 0000.003D.0180.F800.0000.11D4  ;3833 =0 MCT/ALLOW.IB.READ,CALL,J/WAIT7 ; GO WAIT FOR THE RDS BIT TO SET
U 1111. 0000.003C.0180.0800.0000.13B7  ;3834  NOP
U 13B7. 2000.003C.0180.0800.0000.13B8  ;3835  IBC/FLUSH   ; CLEAR THE RDS BIT
U 13B8. 0F00.003C.0180.0A70.0200.13B9  ;3836  ALU_R[0E],VA_ALU,D_0  ; CLEAR FORCE RDS IN CONFIG REG B
U 13B9. 0000.003C.0180.A800.0000.13BA  ;3837  CACHE.P_D[LONG]
U 13BA. 0000.003C.65F0.2C00.0000.13BB  ;3838  Q_ID[SBI.ERR]   ; READ THE ERROR REG
U 13BB. 0810.0038.0180.0970.0000.13BC  ;3839  D_RC[0E]
U 13BC. 001D.2000.0180.0800.0010.13BD  ;3840  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 13BD. 0000.013C.0180.0800.0000.1112  ;3841  Z?    ; BRANCH IF OK
U 1112. 0001.203D.0180.09E8.0000.1149  ;3842 =0 ERROR2,RC[0D]_Q   ; FAILURE NOT DUE TO REFRESH
U 1113. 0000.003C.0180.0800.0000.1114  ;3843 CONT1: NOP
            ;3844
            ;3845 ;////////////////////////////////////////////////////////////////////////////
            ;3846 ;*
            ;3847 ; NOW CHECK THAT THE TIMEOUT BIT CLEARS WITH A FLUSH
            ;3848 ;-
            ;3849
U 1114. 0000.003D.0180.0800.0000.115C  ;3850 =0 SUBTEST
U 1115. 0000.003C.0180.0800.0000.1116  ;3851  NOP
U 1116. 0000.003D.0180.0800.0000.1146  ;3852 =0 ERLOOP
U 1117. 2000.003C.0180.0A58.4200.13BE  ;3853  ALU_R[0B],FLUSH.IB  ; LOAD THE VIBA WITH NON EXISTANT ADRS
U 13BE. 3000.003C.0180.6000.0000.1118  ;3854  LOAD.IB,IBC/START  ; START THE IB REFERANCE
U 1118. 0000.003D.0180.F800.0000.11D8  ;3855 =0 MCT/ALLOW.IB.READ,CALL,J/WAITFORTIMEOUT ; GO WAIT FOR A TIMEOUT
U 1119. 1000.003C.0180.0800.0000.13BF  ;3856  IBC/STOP   ; STOP THE IB
U 13BF. 2810.0038.0180.0970.0000.13C0  ;3857  IBC/FLUSH,D_RC[0E]  ; CLEAR THE ERROR REGISTER
U 13C0. 0000.003C.65F0.2C00.0000.13C1  ;3858  Q_ID[SBI.ERR]   ; READ THE REGISTER
U 13C1. 001D.2000.0180.0800.0010.13C2  ;3859  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 13C2. 0000.013C.0180.0800.0000.111A  ;3860  Z?    ; BRANCH IF OK
U 111A. 0001.203D.0180.09E8.0000.1149  ;3861 =0 ERROR2,RC[0D]_Q   ; TIMEOUT BIT DID NOT CLEAR.
U 111B. 0810.0038.0180.0970.0000.13C3  ;3862  D_RC[0E]   ; CLEAR THE ERROR REGISTER
U 13C3. 0000.003C.6580.3C00.0000.13C4  ;3863  ID[SBI.ERR]_D   ; ...
U 13C4. 0000.003C.0180.0800.0000.111C  ;3864 END00A: NOP
            ;3865
```

```
            ;3866  .PAGE "TEST 1E6 TIMEOUT ADDRESS REGISTER BITS <31:28>"
            ;3867 ;*************************************************************
            ;3868 ;+*
            ;3869 ;
            ;3870 ; TIMEOUT ADDRESS REGISTER BITS <31:28>
            ;3871 ;
            ;3872 ; TEST DESCRIPTION
            ;3873 ;
            ;3874 ; THIS TEST CHECKS THAT BITS <31:28> IN THE TIMEOUT ADDRESS REGISTER
            ;3875 ; ARE FUNCTIONING PROPERLY.
            ;3876 ;
            ;3877 ; SUBTEST 1 - BITS <31:28>=1000 CURRENT MODE 1 * DISABLE PROTECTION
            ;3878 ;
            ;3879 ; SUBTEST 2 - BITS <31:28>=0110 CURRENT MODE 0 * -DISABLE PROTECTION
            ;3880 ;
            ;3881 ; LOGIC DESCRIPTION
            ;3882 ;
            ;3883 ; THIS TEST CHECKS THE TIMEOUT ADDRESS REGISTER BITS <31:28> ON
            ;3884 ; THE SBH MODULE.
            ;3885 ;
            ;3886 ; ERROR DESCRIPTION
            ;3887 ;
            ;3888 ; DATA: EXPECTED CONTENTS OF BITS <31:28>
            ;3889 ;       RECEIVED CONTENTS OF BITS <31:28>
            ;3890 ;--
            ;3891 ;*************************************************************
            ;3892
            ;3893 =0
U 111C, 0000,003D,0180,0800,0000,1131  ;3894 100B: NEWTST
U 111D, 0818,0038,CD80,0800,0000,13C5  ;3895  D_K[.F0]    ; GENERATE A MASK FOR BITS <31:28>
U 13C5, 0B00,003C,0180,0800,0000,13C6  ;3896  D_D.SWAP
U 13C6, 0001,003C,0180,0A80,0000,13C7  ;3897  R[0]_D      ; SAVE IT
U 13C7, 0818,0038,45C0,0800,0000,13C8  ;3898  D_K[.8000],Q_K[.8000]  ; GENERATE MAINTENANCE REGISTER DATA
U 13C8, 0500,003C,0180,0800,0000,13C9  ;3899  D_D.LEFT
U 13C9, 081D,0030,01C0,0800,0000,13CA  ;3900  D_D.OR.Q,Q_ALU    ; FORCE MISS BITS
U 13CA, 0818,0038,4180,0800,0000,13CB  ;3901  D_K[.80]
U 13CB, 0500,003C,0180,0800,0000,13CC  ;3902  D_D.LEFT    ; FORCE TIMEOUT BIT
U 13CC, 001D,0030,0180,0A88,0000,111E  ;3903  R[1]_D.OR.Q    ; SAVE IN R1
            ;3904
            ;3905 ;/////////////////////////////////////////////////////////////
            ;3906 ;+
            ;3907 ; CHECK BITS <31:28>=1000 CURRENT MODE 1 AND DISABLE PROTECTION
            ;3908 ;-
            ;3909
U 111E, 0000,003D,0180,0800,0000,115C  ;3910 =0 SUBTEST
U 111F, 0818,0038,4180,0800,0000,13CD  ;3911  D_K[.80]    ; GENERATE THE EXPECTED DATA
U 13CD, 0B00,003C,0180,0800,0000,13CE  ;3912  D_D.SWAP
U 13CE, 0001,003C,0180,09F0,0000,13CF  ;3913  RC[0E]_D    ; SAVE
U 13CF, 0003,003C,0180,0800,0200,13D0  ;3914  VA_ALU,ALU_0(A)    ; LOAD THE VA
U 13D0, 0800,003C,0180,0A08,0000,13D1  ;3915  D_R[1]     ; GET MAINTENANCE REGISTER DATA
U 13D1, 0000,003C,7580,3C00,0000,13D2  ;3916  ID[MAINT]_D    ; LOAD THE MAINTENANCE REG
U 13D2, 0818,0038,0980,0800,0000,13D3  ;3917  D_K[.2]     ; GENERATE THE PSL DATA
U 13D3, 0B00,003C,0180,0800,0000,13D4  ;3918  D_D.SWAP
U 13D4, 0000,003C,3D80,3C00,0000,1120  ;3919  ID[PSL]_D    ; SET CURRENT MODE 1
U 1120, 0000,003D,0180,0800,0000,1146  ;3920 =0 ERLOOP
```

```
U 1121, 0000.003C.8980.0800.0084.73D5  ;3921  SC_K[.D]    ; SET THE TIMEOUT TRAP FLAG
U 13D5, 0003.001C.0180.0AF8.0000.13D6  ;3922  R[0F]_NOT.MASK
U 13D6, 0000.003C.8580.0800.0084.73D7  ;3923  SC_K[.C]    ; CLEAR THE TIMEOUT BIT
U 13D7, 0803.001C.0180.0800.0000.13D8  ;3924  D_NOT.MASK
U 13D8, 0000.003C.6580.3C00.0000.13D9  ;3925  ID[SBI.ERR]_D
U 13D9, 0000.003C.0180.C800.0000.13DA  ;3926  MCT/READ.P    ; EXECUTE THE MEMORY FUNCTION
U 13DA, 0810.0038.69F0.2D70.0000.13DB  ;3927  Q_ID[TIME.ADDR],D_RC[0E] ; READ THE REGISTER
U 13DB, 001C.0034.01C0.0A00.0000.13DC  ;3928  Q_Q.AND.R[0]    ; MASK
U 13DC, 001D.2000.0180.0800.0010.13DD  ;3929  ALU_Q-D,CLK.UBCC  ; CHECK
U 13DD, 0000.013C.0180.0800.0000.1122  ;3930  Z?
U 1122, 0001.203D.0180.09E8.0000.1149  ;3931 =0 ERROR2,RC[0D]_Q    ; REGISTER INCORRECT
U 1123, 0000.003C.0180.0800.0000.1124  ;3932  NOP
            ;3933
            ;3934 ;////////////////////////////////////////////////////////////////////////
            ;3935 ;*
            ;3936 ; CHECK TIMEOUT ADDRESS REGISTER BITS <31:28>=0110
            ;3937 ;-
            ;3938
U 1124, 0000.003D.0180.0800.0000.115C  ;3939 =0 SUBTEST
U 1125, 0818.0038.A580.0800.0000.13DE  ;3940  D_K[.60]    ; GENERATE THE EXPECTED DATA
U 13DE, 0B00.003C.0180.0800.0000.13DF  ;3941  D_D.SWAP
U 13DF, 0001.003C.0180.09F0.0000.1126  ;3942  RC[0E]_D    ; SAVE
U 1126, 0000.003D.0180.0800.0000.1146  ;3943 =0 ERLOOP
U 1127, 0818.0038.0580.0800.0000.13E0  ;3944  D_K[.1]     ; GENERATE DATA TO SET CURRENT MODE 0
U 13E0, 0B00.003C.0180.0800.0000.13E1  ;3945  D_D.SWAP
U 13E1, 0000.003C.3D80.3C00.0000.13E2  ;3946  ID[PSL]_D    ; SET CURRENT MODE 0
U 13E2, 0800.003C.0180.0A08.0000.13E3  ;3947  D_R[1]     ; GET MAINTENANCE REG DATA
U 13E3, 0000.003C.7580.3C00.0000.13E4  ;3948  ID[MAINT]_D    ; LOAD THE MAINTENANCE REG
U 13E4, 0000.003C.8980.0800.0084.73E5  ;3949  SC_K[.D]    ; SET THE TIMEOUT TRAP FLAG
U 13E5, 0003.001C.0180.0AF8.0000.13E6  ;3950  R[0F]_NOT.MASK
U 13E6, 0000.003C.8580.0800.0084.73E7  ;3951  SC_K[.C]    ; CLEAR THE TIMEOUT BIT
U 13E7, 0803.001C.0180.0800.0000.13E8  ;3952  D_NOT.MASK
U 13E8, 0000.003C.6580.3C00.0000.13E9  ;3953  ID[SBI.ERR]_D
U 13E9, 0000.003C.0180.4000.0000.13EA  ;3954  MCT/READ.V.RCHK    ; EXECUTE THE FUNCTION
U 13EA, 0810.0038.69F0.2D70.0000.13EB  ;3955  Q_ID[TIME.ADDR],D_RC[0E] ; READ THE REGISTER
U 13EB, 001C.0034.01C0.0A00.0000.13EC  ;3956  Q_Q.AND.R[0]    ; MASK
U 13EC, 001D.2000.0180.0800.0010.13ED  ;3957  ALU_Q-D,CLK.UBCC  ; CHECK
U 13ED, 0000.013C.0180.0800.0000.1128  ;3958  Z?
U 1128, 0001.203D.0180.09E8.0000.1149  ;3959 =0 ERROR2,RC[0D]_Q    ; REGISTER INCORRECT
U 1129, 0000.003C.0180.0800.0000.112A  ;3960  NOP
            ;3961
            ;3962
```

```
          ;3963  .PAGE "TEST 1E7 RESERVED"
          ;3964 =0
U 112A, 0000.003D.0180.0800.0000.1131  ;3965 DUMMY1: NEWTST
U 112B, 0000.003C.0180.0800.0000.112C  ;3966  NOP
```

```
            ;3967  .PAGE "TEST 1E8 RESERVED"
            ;3968 =0
U 112C, 0000,003D,0180,0800,0000,1131  ;3969 DUMMY2: NEWTST
U 112D, 0000,003C,0180,0800,0000,13EE  ;3970  NOP
            ;3971
```

```
            ;3972  .PAGE "DUMMY NEWTST"
U 13EE, 0000,003D,0180,0800,0000,1131  ;3973 ENDTST: NEWTST
```

: Location / Line Number Index

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1921: 1890 1892= 1911= 2764= 2765= 2766= 1891
U 1008  2568= 2569= 2570= 2571= 2572= 2573= 2574= 2575=
U 1010  2774= 2775= 2777= 1914 2781= 2782= 2783= 1915
U 1018  2787= 2790= 2792= 2793= 1912= 1913= 1929= 1930=
U 1020  1984= 1986= 1997= 1998= 2382= 2384= 2387= 2389=
U 1028  2392= 2394= 2416= 2417= 2418= 2419= 2428= 2429=
U 1030  2430= 2431= 2501= 2502= 2544= 2545= 2546= 2548=
U 1038  2556= 2557= 2558= 2560= 2563= 2564= 2606= 2607=
U 1040  2608= 2609= 2616= 2617= 2623= 2625= 2628= 2629=
U 1048  2648= 2649= 2651= 2652= 2655= 2657= 2003: 1916
U 1050  2768= 2769= 2833= 2834= 2866= 2867= 2870= 2871=
U 1058  2882= 2883= 2007: 1917 2911= 2912= 2011: 1918
U 1060  2970= 2971= 2973= 2974= 3006= 3007= 3057= 3058=
U 1068  3059= 3060= 3061= 3062= 3065= 3066= 3067= 3068=
U 1070  3073= 3074= 3078= 3084= 3087= 3093= 3094= 3095=
U 1078  3098= 3104= 3111= 3112= 3152= 3153= 3154= 3155=
U 1080  3156= 3157= 3159= 3160= 3175= 3181= 3182= 3183=
U 1088  3186= 3187= 3228= 3229= 3230= 3231= 3232= 3233=
U 1090  3234= 3235= 3239= 3240= 3245= 3251= 3256= 3262=
U 1098  3268= 3274= 3281= 3287= 3288= 3289= 3292= 3293=
U 10A0  3326= 3327= 3328= 3329= 3330= 3331= 3332= 3333=
U 10A8  3343= 3344= 3347= 3353= 3357= 3358= 3390= 3391=
U 10B0  3392= 3393= 3394= 3395= 3396= 3397= 3406= 3407=
U 10B8  3412= 3418= 3419= 3420= 3423= 3424= 3451= 3452=
U 10C0  3453= 3454= 3464= 3465= 3469= 3470= 3481= 3482=
U 10C8  3510= 3511= 3512= 3513= 3525= 3526= 3536= 3537=
U 10D0  3564= 3565= 3566= 3567= 3575= 3576= 3586= 3587=
U 10D8  3614= 3615= 3616= 3617= 3625= 3626= 3636= 3637=
U 10E0  3676= 3677= 3678= 3679= 3680= 3681= 3684= 3685=
U 10E8  3696= 3697= 3705= 3706= 3713= 3719= 3730= 3731=
U 10F0  3747= 3753= 3762= 3768= 3769= 3770= 3773= 3774=
U 10F8  3805= 3806= 3807= 3808= 3812= 3813= 3814= 3815=
U 1100  1876: 1877: 1878: 1879: 1880: 1881: 1882: 1883:
U 1108  1884: 1919 3824= 3825= 1885: 1886: 1887: 1888:
U 1110  3833= 3834= 3842= 3843= 3850= 3851= 3852= 3853=
U 1118  3855= 3856= 3861= 3862= 3894= 3895= 3910= 3911=
U 1120  3920= 3921= 3931= 3932= 3939= 3940= 3943= 3944=
U 1128  3959= 3960= 3965= 3966= 3969= 3970= 1920 1927
U 1130  1928 1973 1974 1975 1976 1977 1978 1979
U 1138  1980 1981 1982 1983 1987 1988 1989 1990
U 1140  1991 1992 1993 1994 1995 1996 2000 2001
U 1148  2002 2005 2006 2014 2016 2055 2056 2057
U 1150  2058 2074 2075 2076 2077 2098: 2078 2080
U 1158  2081 2082 2083 2085 2095 2096 2108 2377
U 1160  2378 2379 2380 2381 2385 2386 2390 2391
U 1168  2395 2420 2421 2425 2427 2432 2433 2434
U 1170  2435 2436 2437 2438 2439 2440 2441 2442
U 1178  2443 2444 2445 2446 2447 2448 2476 2477
U 1180  2478 2503 2504 2505 2506 2549 2550 2551
U 1188  2552 2561 2562 2566 2567 2580 2581 2583
U 1190  2584 2585 2587 2592 2593 2594 2596 2602
U 1198  2604 2605 2613 2614 2618 2619 2620 2621
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2622 2630 2631 2632 2634 2635 2636 2637
U 11A8   2638 2639 2644 2646 2647 2650 2654 2681
U 11B0   2682 2683 2684 2713 2714 2716 2717 2718
U 11B8   2762 2763 2794 2822 2823 2824 2825 2826
U 11C0   2827 2828 2829 2831 2832 2862 2864 2865
U 11C8   2869 2873 2874 2875 2876 2878 2879 2880
U 11D0   2881 2884 2885 2886 2896 2897 2898 2899
U 11D8   2907 2908 2909 2910 2958 2959 2960 2961
U 11E0   2962 2963 2966 2968 2969 3001 3063 3064
U 11E8   3069 3070 3071 3072 3075 3076 3085 3096
U 11F0   3105 3106 3107 3108 3109 3158 3161 3162
U 11F8   3164 3165 3166 3167 3168 3169 3170 3171
U 1200   2113: 2114: 2115: 2116: 2117: 2118: 2119: 2120:
U 1208   2121: 2122: 2123: 2124: 2125: 2126: 2127: 2128:
U 1210   2129: 2130: 2131: 2132: 2133: 2134: 2135: 2136:
U 1218   2137: 2138: 2139: 2140: 2141: 2142: 2143: 2144:
U 1220   2145: 2146: 2147: 2148: 2149: 2150: 2151: 2152:
U 1228   2153: 2154: 2155: 2156: 2157: 2158: 2159: 2160:
U 1230   2161: 2162: 2163: 2164: 2165: 2166: 2167: 2168:
U 1238   2169: 2170: 2171: 2172: 2173: 2174: 2175: 2176:
U 1240   2177: 2178: 2179: 2180: 2181: 2182: 2183: 2184:
U 1248   2185: 2186: 2187: 2188: 2189: 2190: 2191: 2192:
U 1250   2193: 2194: 2195: 2196: 2197: 2198: 2199: 2200:
U 1258   2201: 2202: 2203: 2204: 2205: 2206: 2207: 2208:
U 1260   2209: 2210: 2211: 2212: 2213: 2214: 2215: 2216:
U 1268   2217: 2218: 2219: 2220: 2221: 2222: 2223: 2224:
U 1270   2225: 2226: 2227: 2228: 2229: 2230: 2231: 2232:
U 1278   2233: 2234: 2235: 2236: 2237: 2238: 2239: 2240:
U 1280   2241: 2242: 2243: 2244: 2245: 2246: 2247: 2248:
U 1288   2249: 2250: 2251: 2252: 2253: 2254: 2255: 2256:
U 1290   2257: 2258: 2259: 2260: 2261: 2262: 2263: 2264:
U 1298   2265: 2266: 2267: 2268: 2269: 2270: 2271: 2272:
U 12A0   2273: 2274: 2275: 2276: 2277: 2278: 2279: 2280:
U 12A8   2281: 2282: 2283: 2284: 2285: 2286: 2287: 2288:
U 12B0   2289: 2290: 2291: 2292: 2293: 2294: 2295: 2296:
U 12B8   2297: 2298: 2299: 2300: 2301: 2302: 2303: 2304:
U 12C0   2305: 2306: 2307: 2308: 2309: 2310: 2311: 2312:
U 12C8   2313: 2314: 2315: 2316: 2317: 2318: 2319: 2320:
U 12D0   2321: 2322: 2323: 2324: 2325: 2326: 2327: 2328:
U 12D8   2329: 2330: 2331: 2332: 2333: 2334: 2335: 2336:
U 12E0   2337: 2338: 2339: 2340: 2341: 2342: 2343: 2344:
U 12E8   2345: 2346: 2347: 2348: 2349: 2350: 2351: 2352:
U 12F0   2353: 2354: 2355: 2356: 2357: 2358: 2359: 2360:
U 12F8   2361: 2362: 2363: 2364: 2365: 2366: 2367: 2368:
U 1300   3172 3173 3184 3236 3237 3238 3241 3242
U 1308   3243 3252 3253 3254 3263 3264 3265 3266
U 1310   3275 3276 3277 3278 3279 3290 3334 3335
U 1318   3337 3338 3339 3340 3341 3342 3345 3354
U 1320   3355 3398 3399 3400 3401 3402 3403 3404
U 1328   3405 3408 3409 3410 3421 3455 3456 3457
U 1330   3458 3459 3460 3461 3462 3463 3466 3467
U 1338   3468 3471 3472 3473 3474 3475 3476 3477
U 1340   3478 3479 3480 3514 3515 3516 3517 3518
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3519 3520 3521 3522 3523 3524 3527 3528
U 1350  3529 3530 3531 3532 3533 3534 3535 3568
U 1358  3569 3570 3571 3572 3573 3574 3577 3578
U 1360  3579 3580 3581 3582 3583 3584 3585 3618
U 1368  3619 3620 3621 3622 3623 3624 3627 3628
U 1370  3629 3630 3631 3632 3633 3634 3635 3682
U 1378  3683 3686 3687 3688 3689 3690 3691 3692
U 1380  3694 3695 3698 3700 3701 3702 3703 3704
U 1388  3707 3708 3709 3710 3711 3720 3721 3722
U 1390  3723 3724 3725 3726 3727 3728 3732 3738
U 1398  3739 3740 3741 3742 3743 3744 3745 3754
U 13A0  3755 3756 3757 3758 3759 3760 3771 3809
U 13A8  3810 3811 3816 3817 3818 3820 3821 3822
U 13B0  3823 3826 3828 3829 3830 3831 3832 3835
U 13B8  3836 3837 3838 3839 3840 3841 3854 3857
U 13C0  3858 3859 3860 3863 3864 3896 3897 3898
U 13C8  3899 3900 3901 3902 3903 3912 3913 3914
U 13D0  3915 3916 3917 3918 3919 3922 3923 3924
U 13D8  3925 3926 3927 3928 3929 3930 3941 3942
U 13E0  3945 3946 3947 3948 3949 3950 3951 3952
U 13E8  3953 3954 3955 3956 3957 3958 3973
U 13F0  2021: 2022: 2023: 2024: 2025: 2026: 2027: 2028:
U 13F8  2029: 2030: 2031: 2032: 2033: 2034: 2035: 2036:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR50	1023
Used	1023
Remaining	

Total microwords used in memory U: 1023
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR50.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 1698
ESKAR51.MCR MICRO2 1P(00) 29-JUL-85 15

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1849 MICROTRAP HANDLER
; 1940 Micro Monitor Interface Routines
; 1941   Newtest Routine
; 2011   Disable Cache Routine
; 2210   Set Trap Mask
; 2239   Initialize the SBI
; 2267   Find MS780-E Memory
; 2412   Generate IO Address
; 2438   Set Starting Address
; 2472   CONFIGURE MS780-E/H
; 2548   Select Controller
; 2587   ENABLE NEXT MS780-E
; 2760   WAIT REFRESH SUBROUTINE
; 2823   Wait Loop Routine
; 2854 TEST 1E9 AUTO REFILL HIT AND CP READ HIT
; 2959 TEST 1EA AUTO REFILL MISS AND CP READ MISS
; 3010 TEST 1EB AUTO REFILL AND ROM ENABLE 2 (READ.V.NEWPC)
; 3057 TEST 1EC AUTO REFILL HIT AND CPU WRITE
; 3107 TEST 1ED AUTO REFILL AND CP VALIDATE
; 3156 TEST 1EE AUTO REFILL HIT WITH DISABLE SBI AND ISR
; 3207 TEST 1EF AUTO REFILL HIT AND IB FLUSH
; 3265 TEST 1F0 STALL<=TO TRP ENBLE*BUFFER FULL*DISABLE SBI*-AR OR SBI
; 3323 TEST 1F1 STALL DURING AN INTERRUPT SUMMARY READ
; 3366 TEST 1F2 CPU WRITE MISS AND DATA MEMORY PARITY ERROR
; 3506 TEST 1F3 CPU WRITE HIT AND CACHE DATA PARITY ERROR
; 3625 TEST 1F4 MASK ON INSTRUCTION BUFFER READS
; 3674 TEST 1F5 AUTO REFILL CACNELED BY INSTRUCTION BUFFER FLUSH
; 3728 TEST 1F6 RESERVED
; 3733 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        :1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR51
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; 2.1 Joe Zagarella 11-jul-1985
;1836 ;
;1837 ;  With ms780-e/h configured with upper controller
;1838 ;  only, all tests in this section would fail at
;1839 ;  subroutine "configure ms780-e/h" because this
;1840 ;  subroutine only checked for the lower controller.
;1841 ;  Added code at label "try.upper" to check for upper
;1842 ;  controller.
;1843 ;
;1844 ;
;1845 ;
;1846 ;*****************************************************************************
;1847
```

```
            ;1848  .PAGE
            ;1849  .TOC " MICROTRAP HANDLER"
            ;1850 ;*
            ;1851 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1852 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
            ;1853 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1854 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1855 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1856 ;
            ;1857 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1858 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1859 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1860 ; R[0F] AND DO A RETURN0.
            ;1861 ;
            ;1862 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1863 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1864 ; TO THE CONSOLE.
            ;1865 ;
            ;1866 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
            ;1867 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
            ;1868 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
            ;1869 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
            ;1870 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
            ;1871 ;-
            ;1872
U 1100. 0000.003C.19C0.0A78.0084.7001  ;1873 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101. 0000.003C.05C0.0A78.0084.7001  ;1874 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.09C0.0A78.0084.7001  ;1875 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0DC0.0A78.0084.7001  ;1876 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.11C0.0A78.0084.7001  ;1877 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D5C0.0A78.0084.7001  ;1878 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D9C0.0A78.0084.7001  ;1879 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5DC0.0A78.0084.7001  ;1880 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.01C0.0A78.0084.7001  ;1881 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.85C0.0A78.0084.7001  ;1882 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D. 0000.003C.89C0.0A78.0084.7001  ;1883 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E. 0000.003C.65C0.0A78.0084.7001  ;1884 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.61C0.0A78.0084.7114  ;1885 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
            ;1886
U 1001. 0001.2024.0180.0800.0010.1007  ;1887 TRPH0: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1007. 0000.013C.0180.0800.0000.1002  ;1888  Z?    ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1889 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1890
            ;1891
            ;1892 ;*
            ;1893 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1894 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1895 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1896 ;
            ;1897 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1898 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1899 ;       FAULT STATUS REGISTER
            ;1900 ;       SBI ERROR REGISTER
            ;1901 ;       TIMEOUT ADDRESS REGISTER
            ;1902 ;       MAINTENANCE REGISTER
```

```
            ;1903 ;       CACHE PARITY REGISTER
            ;1904 ;       TB ERROR REGISTER 1
            ;1905 ;       TB ERROR REGISTER 0
            ;1906 ;-
            ;1907
U 1003. 0018.0038.1D80.09E8.0000.1028  ;1908 TRPH1: RC[0D]_SC
U 1028. 0000.003D.D980.0800.0084.714B  ;1909 =0 SETRCF[.9]
U 1029. 0000.003C.49F0.2C00.0000.1013  ;1910  Q_ID[TBER0]
U 1013. 0001.203C.4DF0.2DB0.0000.1017  ;1911  RC[6]_Q,Q_ID[TBER1]
U 1017. 0001.203C.65F0.2DB8.0000.1027  ;1912  RC[7]_Q,Q_ID[SBI.ERR]
U 1027. 0001.203C.6DF0.2DD8.0000.104F  ;1913  RC[0B]_Q,Q_ID[FAULT]
U 104F. 0001.203C.75F0.2DE0.0000.105B  ;1914  RC[0C]_Q,Q_ID[MAINT]
U 105B. 0001.203C.79F0.2DC8.0000.105F  ;1915  RC[9]_Q,Q_ID[PARITY]
U 105F. 0001.203C.69F0.2DC0.0000.1109  ;1916  RC[8]_Q,Q_ID[TIME.ADDR]
U 1109. 0001.203C.81F0.2DD0.0000.1000  ;1917  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.112C  ;1918 1000: RC[0E]_Q,ERROR1
            ;1919
            ;1920 ;*
            ;1921 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
            ;1922 ;-
            ;1923
U 1114. 0001.2024.0180.0800.0010.1115  ;1924 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 1115. 0000.013C.0180.0800.0000.102A  ;1925  Z?     ; BRANCH IF NO
U 102A. 0001.2036.0180.0AF8.0000.0001  ;1926 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1 ; RETURN
U 102B. 0000.003C.0180.0800.0000.1003  ;1927  J/TRPH1     ; REPORT UNEXPECTED TRAP
            ;1928
            ;1929
            ;1930 ;*
            ;1931 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1932 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1933 ;  NEWTST
            ;1934 ;  ERLOOP
            ;1935 ;  ERROR1
            ;1936 ;  ERROR2
            ;1937 ;
            ;1938
```

```
                ;1939  .PAGE
                ;1940  .TOC " Micro Monitor Interface Routines"
                ;1941  .TOC "   Newtest Routine"
                ;1942 ;++
                ;1943 ; FUNCTIONAL DESCRIPTION:
                ;1944 ;
                ;1945 ; This routine Initializes the following registers:
                ;1946 ;  PSL    to 1F
                ;1947 ;  CES    to 0
                ;1948 ;  ACC.CS    to disable accellerator
                ;1949 ;  SIR    to 0
                ;1950 ;  CLK.CS    to stop interval timer
                ;1951 ;  TBER0    to disable the TB
                ;1952 ;  SBI.MAINT to    0
                ;1953 ;  SBI.ERR    to 0
                ;1954 ;  FAULT    to 0
                ;1955 ;  COMP    to 0
                ;1956 ;  RC[0E]    to  0
                ;1957 ;  R[0F]    to 0
                ;1958 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1959 ; A SCOPE call is then made to the Monitor.
                ;1960 ;
                ;1961 ; CALLING CONSTRAINT:
                ;1962 ;
                ;1963 ; =0
                ;1964 ;
                ;1965 ; SIDE EFFECTS:
                ;1966 ;
                ;1967 ; D Reg is destroyed
                ;1968 ; SC is destroyed
                ;1969 ;--
U 1116, 0000.003C.65F8.0800.0084.7117  ;1970 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1117, 0818.0038.8D80.0800.0000.1118  ;1971  d_k[.1f]  ; D=1F
U 1118, 0D00.003C.0180.0800.0000.1119  ;1972  d_dal.sc  ; D=001F0000
U 1119, 0000.003C.3D80.3C00.0000.111A  ;1973  id[psl]_d  ; psl = 001F0000
U 111A, 0818.0038.0580.0800.0000.111B  ;1974  d_k[.1]  ; Clear the CES register
U 111B, 0D00.003C.0180.0800.0000.111C  ;1975  d_dal.sc  ; D = 00010000
U 111C, 0F00.003C.3180.3C00.0000.111D  ;1976  id[ces]_d,d_0  ; ces = 00010000
U 111D, 0000.003C.5D80.3C00.0000.111E  ;1977  id[acc.cs]_d  ; acc.cs = 0
U 111E, 0000.003C.3980.3C00.0000.111F  ;1978  id[sir]_d  ; sir = 0
U 111F, 0000.003C.2980.3C00.0000.1120  ;1979  id[clk.cs]_d  ; clk.cs = 0
U 1120, 0000.003C.4980.3C00.0000.102C  ;1980  id[tber0]_d  ; tber0 = 0
U 102C, 0000.003D.0180.0800.0000.114E  ;1981 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1982  intrpt.strobe,  ; Strobe interrupts
U 102D, 0818.0038.C180.0800.4000.1121  ;1983      d_k[.ffff]  ; Setup to clear SBI error register and
U 1121, 0801.0028.6580.3C00.0000.1122  ;1984  id[sbi.err]_d,d_not.d ; and fault register
U 1122, 0F00.003C.6D80.3C00.0000.1123  ;1985  id[fault]_d,d_0  ; ...
U 1123, 0000.003C.6D80.3C00.0000.1124  ;1986  id[fault]_d  ; fault = 0
U 1124, 0000.003C.7580.3C00.0000.1125  ;1987  id[maint]_d  ; MAINT = 0
U 1125, 0000.003C.6580.3C00.0000.1126  ;1988  id[sbi.err]_d  ; sbi error reg = 0
U 1126, 0000.003C.7180.3C00.0000.1127  ;1989  id[comp]_d  ; comp reg = 0
U 1127, 0000.003C.E580.3C00.0000.1128  ;1990  id[T9]_d  ; Clear code for subroutine START.TEST
U 1128, 0001.003C.0180.09F0.0000.1129  ;1991  rc[0e]_d  ;
U 1129, 0001.003C.0180.0AF8.0000.112A  ;1992  r[0f]_d  ; Clear expected trap mask
U 112A, 0001.003C.0180.09F8.0000.102E  ;1993  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 102E, 0000,003D,0180,0800,0000,1130  ;1994 =0 call[disable.cache] ; Disable the cache
U 102F, 0F03,003C,0180,09E8,0000,105E  ;1995  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1996
U 112B, 0818,0038,0980,0800,0000,105E  ;1997 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 112C, 0810,0038,0180,0978,0000,112D  ;1998 ERROR1: D_RC[0F]
U 112D, 0819,0034,4D80,0800,0000,104E  ;1999  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2000 104E: D_D.OR.K[.4],J/CALLCON
          ;2001
U 112E, 0810,0038,0180,0978,0000,112F  ;2002 ERROR2: D_RC[0F]
U 112F, 0819,0034,4D80,0800,0000,105A  ;2003  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2004 105A: D_D.OR.K[.6],J/CALLCON
          ;2005
          ;2006 105E:
          ;2007 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2008  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
          ;2009
```

```
            ;2010  .PAGE
            ;2011  .TOC "   Disable Cache Routine"
            ;2012 ;**
            ;2013 ; FUNCTIONAL DESCRIPTION:
            ;2014 ;
            ;2015 ; This routine disables cache hits by setting bits <16:15>
            ;2016 ; in the maintenance register.
            ;2017 ;
            ;2018 ; CALLING SEQUENCE:
            ;2019 ;
            ;2020 ; =0
            ;2021 ;
            ;2022 ; SIDE EFFECTS:
            ;2023 ;
            ;2024 ; D & Q Registers destroyed
            ;2025 ;--
            ;2026 DISABLE.CACHE:
U 1130. 0818.0038.4580.0800.0000.1131  ;2027  d_k[.8000]  ; ... and seed constant
U 1131. 0500.003C.0180.0800.0000.1132  ;2028  d_d.left  ; Generate final constant
U 1132. 0819.0030.4580.0800.0000.1133  ;2029  d_d.or.k[.8000]  ; ... = 00018000
U 1133. 0000.003E.7580.3C00.0000.0001  ;2030  id[maint]_d,return1 ; Disable cache and return
            ;2031
            ;2032 ;*
            ;2033 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2034 ;-
            ;2035
U 13F0. 0000.003C.0180.0800.0000.13F1  ;2036 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;2037 13F1: NOP
U 13F2. 0000.003C.0180.0800.0000.13F3  ;2038 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;2039 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;2040 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;2041 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;2042 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;2043 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;2044 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;2045 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;2046 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;2047 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;2048 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;2049 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;2050 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1134  ;2051 13FF: NOP
            ;2052
            ;2053
            ;2054 ;*
            ;2055 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2056 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2057 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2058 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2059 ;
            ;2060 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2061 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2062 ;
            ;2063 ;  =0 D_0,CALL,J/DC5
            ;2064 ;   NOP
```

```
            ;2065 ;  =0 CALL,J/DC5
            ;2066 ;-
            ;2067
U 1134, 0018.0038.1980.0800.0000.1144  ;2068 DC0: ALU_0(K),J/S00
U 1135, 0018.0038.1980.0800.0000.1145  ;2069 DC1: ALU_0(K),J/S01
U 1136, 0018.0038.1980.0800.0000.1146  ;2070 DC2: ALU_0(K),J/S10
U 1137, 0018.0038.1980.0800.0000.1147  ;2071 DC3: ALU_0(K),J/S11
U 1138, 0018.0038.0980.0800.0000.1144  ;2072 DC4: ALU_K[.2],J/S00
U 1139, 0018.0038.0980.0800.0000.1145  ;2073 DC5: ALU_K[.2],J/S01
U 113A, 0018.0038.0980.0800.0000.1146  ;2074 DC6: ALU_K[.2],J/S10
U 113B, 0018.0038.0980.0800.0000.1147  ;2075 DC7: ALU_K[.2],J/S11
U 113C, 0018.0038.0580.0800.0000.1144  ;2076 DC8: ALU_K[.1],J/S00
U 113D, 0018.0038.0580.0800.0000.1145  ;2077 DC9: ALU_K[.1],J/S01
U 113E, 0018.0038.0580.0800.0000.1146  ;2078 DCA: ALU_K[.1],J/S10
U 113F, 0018.0038.0580.0800.0000.1147  ;2079 DCB: ALU_K[.1],J/S11
U 1140, 0018.0038.C180.0800.0000.1144  ;2080 DCC: ALU_K[.FFFF],J/S00
U 1141, 0018.0038.C180.0800.0000.1145  ;2081 DCD: ALU_K[.FFFF],J/S01
U 1142, 0018.0038.C180.0800.0000.1146  ;2082 DCE: ALU_K[.FFFF],J/S10
U 1143, 0018.0038.C180.0800.0000.1147  ;2083 DCF: ALU_K[.FFFF],J/S11
            ;2084
U 1144, 0118.0038.1B80.0800.0000.1148  ;2085 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1145, 0118.0038.0B80.080C.0000.1148  ;2086 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 1146, 0118.0038.0780.0800.0000.1148  ;2087 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 1147, 0118.0038.C380.0800.0000.1148  ;2088 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
            ;2089
U 114B, 0100.003E.0380.0800.0000.0001  ;2090 SX: D_D.LEFT2,SI/7,RETURN1  ; EXIT
            ;2091
            ;2092
            ;2093 ;*
            ;2094 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2095 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2096 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2097 ; TYPING IT.
            ;2098 ;-
            ;2099
U 1149, 0810.0038.4D80.0978.0000.114A  ;2100 SUBTST: D_RC[0F],KMX/.FF00
U 114A, 0019.4016.4D80.09F8.0000.0001  ;2101  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2102
            ;2103 ;*
            ;2104 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2105 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2106 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2107 ;-
            ;2108
U 114B, 0010.0038.01C0.0978.0000.114C  ;2109 SETRCF: Q_RC[0F]
U 114C, 0019.2034.4DC0.0800.0000.114D  ;2110  Q_Q.AND.K[.FF00]
U 114D, 0019.2032.1D80.09F8.0000.0001  ;2111  RC[0F]_Q.OR.SC,RETURN1
U 1155, 0019.2000.05C0.0800.0000.12AB  ;2112 1155: Q_Q-K[.1],J/12AB
U 12AA, 0000.003C.0180.0800.0000.114E  ;2113 12AA: NOP    ; ECO LOCATION
            ;2114
            ;2115
            ;2116 ;*
            ;2117 ; THIS SUBROUTINE EXECUTE AN SBI UNJAM SEQUENCE
            ;2118 ;-
            ;2119 SBI.UNJAM:
```

```
U 114E, 0F00.003C.0180.0800.0000.114F  ;2120 UNJAM: D_0
U 114F, 0000.003C.7580.3C00.0000.1150  ;2121  ID[MAINT]_D   ; RELOAD THE MAT REG
U 1150, 0F00.003C.0180.8000.0000.1151  ;2122  D_0,MCT/SBI.HOLD   ; ASSERT HOLD FOR 16 CYCLES
U 1151, 0019.0020.6180.8000.0010.1152  ;2123 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD ; DONE YET?
U 1152, 0000.013C.0180.8000.0000.1030  ;2124  Z?,MCT/SBI.HOLD   ; BRANCH IF YES
U 1030, 0819.0014.0580.8000.0000.1151  ;2125 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM0 ; CONTINUE HOLDING THE BUS
          ;2126
U 1031, 0F00.003C.0180.8800.0000.1153  ;2127  D_0,MCT/SBI.HOLD+UNJAM  ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
U 1153, 0019.0020.6180.8800.0010.1154  ;2128 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD+UNJAM ; DONE YET?
U 1154, 0000.013C.0180.8800.0000.1032  ;2129  Z?,MCT/SBI.HOLD+UNJAM  ; BRANCH IF YES
U 1032, 0819.0014.0580.8800.0000.1153  ;2130 =0 D_D+K[.1],MCT/SBI.HOLD+UNJAM,J/UNJAM1 ; CONTINUE UNJAM
          ;2131
U 1033, 0F00.003C.0180.8000.8000.1156  ;2132  D_0,MCT/SBI.HOLD,INTRPT.ACK ; ASSERT HOLD FOR 16 CYCLES
U 1156, 0019.0020.6180.8000.0010.1157  ;2133 UNJAM2: ALU_D[XOR]K[.F],MCT/SBI.HOLD,CLK.UBCC ; DONE YET?
U 1157, 0000.013C.0180.8000.0000.1034  ;2134  Z?,MCT/SBI.HOLD   ; BRANCH IF YES
U 1034, 0819.0014.0580.8000.0000.1156  ;2135 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM2 ; CONTINUE HOLDING
          ;2136
U 1035, 0000.003C.8580.0800.0084.7158  ;2137  SC_K[.C]
U 1158, 0803.001C.0180.0800.4000.1159  ;2138  D_NOT.MASK,INTRPT.STROBE
U 1159, 0000.003E.7580.3C00.0000.0001  ;2139  ID[MAINT]_D,RETURN1  ; TURN OFF THE SBI
          ;2140
          ;2141 ;+
          ;2142 ; THE FOLLOWING SUBROUTINE GENERATES CONSTANTS AND SAVES THEM IN THE RA
          ;2143 ; SCRATCH PADS. FOLLOWING ARE THE CONSTANTS GENERATED:
          ;2144 ;
          ;2145 ; R[2] = FORCE REPLACE GROUP 1 AND FORCE MISS GROUP 0
          ;2146 ; R[3] = R[6] PLUS DISABLE SBI CYCLE
          ;2147 ; R[4] = CPU ADDRESS FOR AUTO REFILL (0)
          ;2148 ; R[5] = IB ADDRESS FOR AUTO REFILL (3FC)
          ;2149 ; R[6] = FORCE REPLACE GROUP 0 AND FORCE MISS GROUP 1
          ;2150 ; R[7] = GROUP 0 AND GROUP 1 MATCH BIT MASK
          ;2151 ; R[8] = GROUP 0 MATCH BIT
          ;2152 ; R[9] = GROUP 1 MATCH BIT
          ;2153 ; R[A] = R[6] PLUS REVERSE PARITY GROUP 0, BYTE 0 (DATA)
          ;2154 ; R[B] = TIMEOUT ADDRESS ( = 01000000 )
          ;2155 ; R[C] = SILO COUNTER MASK
          ;2156 ; R[D] = ADDRESS TO FORCE RDS (4020)
          ;2157 ; R[E] = ADDRESS OF MEMORY CONFIGURATION REGISTER B
          ;2158 ;-
          ;2159
          ;2160 =0
          ;2161 MAINTREGSETUP:
U 1036, 0000.003D.F580.0800.0084.714B  ;2162  SETRCF[.A]   ; Set the Argument count to CONSOLE
U 1037, 0000.003C.0180.0800.0000.1038  ;2163  NOP
U 1038, 0000.003D.0180.0800.0000.11B2  ;2164 =0 CALL[CONFIG.MS780E]  ; Find MS780E memory and configure it
U 1039, 0810.0038.0180.0970.0000.115A  ;2165  D_RC[0E]   ; Get CNFG Reg A Address
U 115A, 0019.0014.1180.0AF0.0000.115B  ;2166  R[0E]_D+K[.4]   ; Make it in CNFG B Address
U 115B, 0818.0038.8580.0800.0000.115C  ;2167  D_K[.C]    ; Form "FORCE RDS" constant for
          ;2168       ; ...CNFG Reg B in R[0D]
          ;2169       ; ...First ECC substitute bits for
          ;2170       ; ...data of zero
U 115C, 0000.003C.F580.0800.0084.715D  ;2171  SC_K[.A]   ; And now the Diagnostic Mode bits
          ;2172  D_D.ORNOT.MASK,   ; ...and save them in
U 115D, 0801.001C.0180.0AE8.0000.103A  ;2173   R[0D]_ALU  ; ...R[0D]
U 103A, 0000.003D.0180.0800.0000.114E  ;2174 =0 CALL[SBI.UNJAM]   ;
```

```
U 103B. 0003,003C,0180,0AA0,0000,103C  ;2175  R[4]_0
U 103C. 0818,0039,5580,0800,0000,1140  ;2176 =0 D_K[.3F],CALL,J/DCC  ; GENERATE 3FC
U 103D. 0001,003C,0180,0AA8,0000,103E  ;2177  R[5]_D    ; SAVE
U 103E. 0818,0039,7580,0800,0000,1134  ;2178 =0 D_K[.20],CALL,J/DC0  ; GENERATE 200(X)
U 103F. 0001,003C,0180,0AC0,0000,115E  ;2179  R[8]_D    ; GROUP 0 MATCH BIT
U 115E. 0501,003C,0180,0AB8,0000,115F  ;2180  R[7]_D,D_D.LEFT   ; PART OF MASK
U 115F. 0001,003C,0180,0AC8,0000,1160  ;2181  R[9]_D    ; GROUP 1 MATCH BIT
U 1160. 0000,003C,0180,0A38,0000,1161  ;2182  LAB_R[7]
U 1161. 001C,2030,0180,0AB8,0000,1162  ;2183  R[7]_LA.OR.D   ; COMPLETE THE MASK
U 1162. 0100,003C,0180,0800,0000,1163  ;2184  D_D.LEFT2
U 1163. 0500,003C,0180,0800,0000,1164  ;2185  D_D.LEFT   ; GENERATE FR GR0 AND FM GR1
U 1164. 0501,003C,0180,0A90,0000,1165  ;2186  R[2]_D,D_D.LEFT
U 1165. 0500,003C,01E0,0800,0000,1166  ;2187  Q_D,D_D.LEFT   ; ...
U 1166. 081D,0030,01E0,0AB0,0000,1167  ;2188  R[6]_D.OR.Q,D_ALU,Q_D  ; SAVE
U 1167. 0000,003C,7580,3C00,0000,1168  ;2189  ID[MAINT]_D   ; SETUP THE MAINTENANCE REGISTER
U 1168. 0000,003C,8580,0800,0084,7169  ;2190  SC_K[.C]   ; SETUP TO GENERATE BIT 12
U 1169. 0801,001C,0180,0800,0000,116A  ;2191  D_D.ORNOT.MASK   ; INSERT BIT 12
U 116A. 0001,003C,0180,0A98,0000,116B  ;2192  R[3]_D    ; SAVE
U 116B. 0C00,003C,0180,0A10,0000,116C  ;2193  LAB_R[2],D_Q   ; COMPLETE DATA FOR R2
U 116C. 0500,003C,0180,0800,0000,116D  ;2194  D_D.LEFT
U 116D. 001C,2030,0180,0A90,0000,116E  ;2195  R[2]_LA.OR.D   ;
U 116E. 0818,0038,5180,0800,0000,116F  ;2196  D_K[.1E]   ; GENERATE CONSTANT FOR R[A]
U 116F. 0000,003C,65F8,0800,0084,7170  ;2197  SC_K[.10],Q_0   ; ...
U 1170. 0D00,003C,01C0,0A30,0000,1171  ;2198  D_DAL.SC,Q_R[6]   ; ...
U 1171. 001D,0030,0180,0AD0,0000,1172  ;2199  R[0A]_D.OR.Q   ; SAVE
U 1172. 0818,0038,0580,0800,0000,1173  ;2200  D_K[.1]    ; GENERATE A TIMEOUT ADDRESS
U 1173. 0B00,003C,0180,0800,0000,1174  ;2201  D_D.SWAP   ; ...
U 1174. 0001,003C,0180,0AD8,0000,1175  ;2202  R[0B]_D    ; SAVE
U 1175. 0818,0038,61F8,0800,0000,1176  ;2203  D_K[.F],Q_0
U 1176. 0D00,003C,0180,0800,0000,1177  ;2204  D_DAL.SC
U 1177. 0001,003C,0180,0AE0,0000,1178  ;2205  R[0C]_D    ; SILO COUNTER MASK
U 1178. 0000,003E,0180,0800,0000,0001  ;2206  RETURN1    ; Return to caller
            ;2207
            ;2208
```

```
            ;2209  .PAGE
            ;2210  .TOC "   Set Trap Mask"
            ;2211 ;**
            ;2212 ; FUNCTIONAL DESCRIPTION:
            ;2213 ;
            ;2214 ; This routine is used to set a bit in the Micro Trap Mask
            ;2215 ; R[0F].
            ;2216 ;
            ;2217 ; CALLING CONSTRAINT:
            ;2218 ;
            ;2219 ; =0
            ;2220 ;
            ;2221 ; INPUTS:
            ;2222 ;
            ;2223 ; SC - Contains bit number to set.
            ;2224 ;
            ;2225 ; OUTPUTS:
            ;2226 ;
            ;2227 ; rc[0E] - Contains the current trap mask
            ;2228 ;
            ;2229 ; SIDE EFFECTS:
            ;2230 ;
            ;2231 ; d regsiter is destroyed
            ;2232 ;--
            ;2233 SET_TRAP_MASK:
U 1179, 0800,003C,0180,0A78,0000,117A  ;2234  d_r[0f]
U 117A, 0801,001C,0180,0AF8,0000,117B  ;2235  r[0f]_d.ornot.mask,d_alu ; Set mask
U 117B, 0001,003E,0180,0AF0,0000,0001  ;2236  r[0E]_d,return1    ; Save mask and return
            ;2237
```

```
            ;2238  .PAGE
            ;2239  .TOC "   Initialize the SBI"
            ;2240 ;++
            ;2241 ; FUNCTIONAL DESCRIPTION:
            ;2242 ;
            ;2243 ; This routine performs the following functions:
            ;2244 ;
            ;2245 ;  Executes an SBI Unjam
            ;2246 ;  Clears the SBI Fault Latch
            ;2247 ;  Clears the SBI Error Register
            ;2248 ;
            ;2249 ; CALLING CONSTRAINT:
            ;2250 ;
            ;2251 ; =0
            ;2252 ;
            ;2253 ; SIDE EFFECTS:
            ;2254 ;
            ;2255 ; D & Q Register destroyed
            ;2256 ;--
            ;2257 =0
            ;2258 SBI.INIT:
U 1040, 0000,003D,0180,0800,0000,114E  ;2259  call[sbi.unjam]  ; Unjam the SBI
U 1041, 0818,0038,C180,0800,0000,117C  ;2260  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 117C, 0801,0028,6580,3C00,0000,117D  ;2261  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 117D, 0F00,003C,6D80,3C00,0000,117E  ;2262  id[fault]_d,d_0
U 117E, 0000,003C,6D80,3C00,0000,117F  ;2263  id[fault]_d
U 117F, 0000,003E,6580,3C00,0000,0001  ;2264  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2265
```

```
;2266  .PAGE
;2267  .TOC "   Find MS780-E Memory"
;2268 ;++
;2269 ; FUNCTIONAL DESCRIPTION:
;2270 ;
;2271 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
;2272 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
;2273 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
;2274 ;     3D will hold the I/O base address of the MS780-E subsystems found
;2275 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
;2276 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
;2277 ;     first MS780-E found will have  its  starting  address  set  to  0
;2278 ;     (zero).
;2279 ;     All the other MS780-E/H's found will have their starting  address
;2280 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
;2281 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
;2282 ;     apropriate  action.  Register RC[0E] is used as a pointer to the
;2283 ;     current MS780-E unit under test.
;2284 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
;2285 ;     set to maximum.
;2286 ;
;2287 ; CALLING CONSTRAINT:
;2288 ;
;2289 ; =00
;2290 ;
;2291 ; OUTPUTS:
;2292 ;
;2293 ; rc[0e] - Contains address of Configuration Register
;2294 ; r[0]  - Contains TR level
;2295 ;
;2296 ; SIDE EFFECTS;
;2297 ;
;2298 ; D register is destroyed.
;2299 ;--
;2300 =0
;2301 FIND.MS780E:
U 1042. 0000.003D.0180.0800.0000.1040  ;2302  call[sbi.init]  ; Make sure SBI is enabled
U 1043. 0000.003C.0180.0800.0000.1044  ;2303  NOP
U 1044. 0000.003D.0180.0800.0000.1130  ;2304 =0 call[disable.cache] ; Make sure cache is disabled and SBI
          ;2305      ; ...is turned on
U 1045. 0003.003C.0180.0988.0000.1180  ;2306  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1180. 0818.0038.1980.0800.0000.1181  ;2307  d_k[zero]  ; Clear MS780-E/H counter
U 1181. 0000.003C.F980.3C00.0000.1182  ;2308  id[3e]_d  ; ...
U 1182. 0018.0038.0580.0A80.0000.1183  ;2309  r[0]_k[.1]  ; Init TR counter
U 1183. 0F03.003C.0180.09F0.0000.1046  ;2310  d_0,rc[0E]_0  ; Clear "Save" location for
          ;2311      ; ...CNFG A Reg
          ;2312 =0
          ;2313 FIND.MEM.LOOP:
U 1046. 0800.003D.0180.0A00.0000.11A9  ;2314  d_r[0],call[generate.io]; Generate address of TR level
U 1047. 0001.003C.0180.09F0.0200.1048  ;2315  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1048. 0000.003D.8980.0800.0084.7179  ;2316 =0 set.trap.mask[.d] ; Enable timeout micro traps
          ;2317  d[long]_cache.p, ; Read the specified tr level
U 1049. 0000.003C.0180.C970.0000.1184  ;2318   ic_rc[0e]  ; ...
U 1184. 0000.003C.0180.0A78.0010.1185  ;2319  alu_r[0f],clk.ubcc ; Check for no response
U 1185. 0001.013C.0180.0880.0082.104A  ;2320  z?,sc_d,ia_ra[0] ; Branch if no adapter here
```

```
U 104A, 0000,003C,0180,0800,0000,1186  ;2321 =0 j/check.adpt.code ; Check for a memory
U 104B, 0000,003C,0180,0800,0000,11A5  ;2322  j/find.mem.lp1  ; Try next tr
          ;2323 CHECK.ADPT.CODE:
U 1186, 0000,003C,1D80,0800,1404,7187  ;2324  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1187, 0000,163C,0180,0800,0000,1008  ;2325  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,7188  ;2326 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,7189  ;2327  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11A5  ;2328  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11A5  ;2329  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,718E  ;2330  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11A5  ;2331  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1192  ;2332  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1192  ;2333  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2334  ;
          ;2335  ; Found an MS780-A or -C. Set the starting address
          ;2336  ; to maximum.
          ;2337  ;
U 1188, 0818,0038,5D80,0800,0000,118A  ;2338 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1189, 0818,0038,0580,0800,0000,118A  ;2339 MS780.C:d_k[.1]  ; Setup bits<16:14> for -C controller
          ;2340 MS780.COMMON:
U 118A, 0819,0038,7180,0800,0000,118B  ;2341  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 118B, 0000,003C,01F8,0803,0080,D18C  ;2342  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 118C, 0D00,003C,0180,0800,0000,118D  ;2343  d_dal.sc  ; Put data in bits<29:14>
          ;2344  cache.p_d[long], ; Set the starting address to maximum
U 118D, 0000,003C,0180,A800,0000,11A5  ;2345       j/find.mem.lp1 ; and continue TR scan
          ;2346  ;
          ;2347  ; Found an MA780. Set the starting address to maximum
          ;2348  ;
          ;2349 MA780.MAX:
U 118E, 0818,0038,39F8,0803,0000,118F  ;2350  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 118F, 0D00,003C,0180,0803,0000,1190  ;2351  d_dal.sc,va_va+4 ; Put in proper position
U 1190, 0000,003C,0180,0803,0000,1191  ;2352  va_va+4  ; Point to Port Invalidate Ctrl Register
          ;2353  cache.p_d[long], ; Set starting address to maximum
U 1191, 0000,003C,0180,A800,0000,11A5  ;2354       j/find.mem.lp1
          ;2355 ;
          ;2356 ; Found an MS780E
          ;2357 ;
          ;2358 FOUND.MS780E:
U 1192, 0000,003C,F9F0,2C00,0000,1193  ;2359  q_id[3e]  ; Set up to see how many MS780-E's
          ;2360  alu_q.xor.k[.3], ; Are there Maximum units?
U 1193, 0019,2020,0D80,0800,0010,1194  ;2361  clk.ubcc ; Check condition codes
U 1194, 0000,013C,0180,0800,0000,104C  ;2362  z?  ; Are we at maximum units for system
U 104C, 0000,003C,0180,0800,0000,1195  ;2363 =0 J/ms780e.tst  ; No go find how many units ther are
U 104D, 0818,0038,C180,0800,0000,1050  ;2364  d_k[.ffff]  ; Yes set address to maximum
U 1050, 0000,003D,0180,0800,0000,11AD  ;2365 =0 call[set.start.addr] ; .....
U 1051, 0000,003C,0180,0800,0000,11A5  ;2366  j/find.mem.lp1  ; Go check to see if we are done
          ;2367 ;
          ;2368 MS780E.TST:
          ;2369  alu_rc[01],  ; Check "Found MS780E Flag"
U 1195, 0010,0038,0180,0908,0010,1196  ;2370  clk.ubcc ; Check condition codes
U 1196, 0810,0138,0180,0970,0000,1052  ;2371  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;2372 =0 j/not.first.ms780e, ; No
U 1052, 0818,0038,C180,0800,0000,1056  ;2373  d_k[.ffff] ; Set starting address
U 1053, 0001,003C,0180,0988,0000,1197  ;2374  rc[01]_d  ; Yes put base address in rc[01]
U 1197, 0818,0038,7980,0800,0000,1198  ;2375  d_k[.30]  ; Set up SC to point to first unit
```

```
U 1198, 0019,0014,F580,0800,0082,1199  ;2376  alu_d+k[.a],sc_alu ; ...
U 1199, 0810,0038,0180,0908,0000,119A  ;2377  d_rc[01]  ; Put base address of unit in D
U 119A, 0000,003C,0180,3400,0000,119B  ;2378  id(sc)_d  ; Put base address in ID pointed to by SC
U 119B, 0F00,003C,0180,0800,0000,1054  ;2379  d_0   ; Prepare to set starting address to Zero
U 1054, 0000,003D,0180,0800,0000,11AD  ;2380 =0 call[set.start.addr] ; Go do it
          ;2381  j/find.mem.lp1.  ; Check to see if we are done
U 1055, 0003,003C,0180,09E8,0000,11A5  ;2382   rc[0d]_0 ; ....
          ;2383 =0
          ;2384 NOT.FIRST.MS780E:
U 1056, 0000,003D,0180,0800,0000,11AD  ;2385  call[set.start.addr] ; Go set starting address to maximum
U 1057, 0000,003C,F9F0,2C00,0000,119C  ;2386  q_id[3e]  ; Get the number of MS780-Es found
U 119C, 0819,2014,0580,0800,0000,119D  ;2387  d_q+k[.1]  ; Increment this counter
U 119D, 0000,003C,F980,3C00,0000,119E  ;2388  id[3e]_d  ; Save the counter
U 119E, 0018,0038,11C0,0800,0000,119F  ;2389  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2390     ; ...were the I/O base addresses will be stored
U 119F, 081D,2000,7980,0800,0000,11A0  ;2391  d_q-d,k[.30]  ; ... adjust pointer
U 11A0, 0819,0014,7980,0800,0000,11A1  ;2392  d_d+k[.30]  ; Point the SC to the ID register
U 11A1, 0000,003C,F580,0800,0000,11A2  ;2393  k[.a]   ; ...to receive I/O base address
U 11A2, 0019,0014,F580,0800,0082,11A3  ;2394  alu_d+k[.a],sc_alu ; ...
U 11A3, 0810,0038,0180,0970,0000,11A4  ;2395  d_rc[0e]  ; Get base address to d
U 11A4, 0000,003C,0180,3400,0000,11A5  ;2396  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2397   ;
          ;2398   ; Try next tr
          ;2399   ;
          ;2400 FIND.MEM.LP1:
U 11A5, 0818,0014,0580,0A80,0000,11A6  ;2401  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2402  alu_d.and.k[.f],
U 11A6, 0019,0034,6180,0800,0010,11A7  ;2403      clk.ubcc  ; Check if all done
U 11A7, 0000,013C,0180,0800,0000,1058  ;2404  z?   ; Branch if yes
U 1058, 0000,003C,0180,0800,0000,1046  ;2405 =0 j/find.mem.loop  ; Try next TR
U 1059, 0810,0038,0180,0908,0010,11A8  ;2406  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11A8, 0000,013C,0180,0800,0000,105C  ;2407  z?   ; Check condition codes
U 105C, 0001,003E,0180,09F0,0000,0002  ;2408 =0 return2,rc[0e]_d ; Yes MS780E was found
U 105D, 0000,003E,0180,0800,0000,0001  ;2409  return1   ; No MS780E found
          ;2410
```

```
            ;2411  .PAGE
            ;2412  .TOC "   Generate IO Address"
            ;2413 ;++
            ;2414 ; FUNCTIONAL DESCRIPTION:
            ;2415 ;
            ;2416 ; This routine is used to generate an SBI Configuration Register
            ;2417 ; address from a TR level.
            ;2418 ;
            ;2419 ; CALLING CONSTRAINT:
            ;2420 ;
            ;2421 ; =0
            ;2422 ;
            ;2423 ; INPUTS:
            ;2424 ;
            ;2425 ; D - Contains TR level
            ;2426 ;
            ;2427 ; OUTPUTS:
            ;2428 ;
            ;2429 ; D - Contains SBI IO address
            ;2430 ;--
            ;2431 GENERATE.IO:
U 11A9, 0000.003C.89F8.0800.0084.71AA  ;2432  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11AA, 0D00.003C.8D80.0800.0084.71AB  ;2433  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11AB, 0000.003C.0980.0800.0084.B1AC  ;2434  sc_sc-k[.2]  ; insert bit 29
U 11AC, 0801.001E.0180.0800.0000.0001  ;2435  d_d.ornot.mask,return1 ; and return
            ;2436
```

```
            ;2437  .PAGE
            ;2438  .TOC "   Set Starting Address"
            ;2439 ;++
            ;2440 ; FUNCTIONAL DESCRIPTION:
            ;2441 ;
            ;2442 ; This routine is used to set the starting address of the
            ;2443 ; memory to the specified value.
            ;2444 ;
            ;2445 ; CALLING CONSTRAINT:
            ;2446 ;
            ;2447 ; =0
            ;2448 ;
            ;2449 ; INPUTS:
            ;2450 ;
            ;2451 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2452 ;     (B Register Image divided by 2**16)
            ;2453 ; rc[0] - Contains Address of Register A
            ;2454 ;
            ;2455 ; OUTPUTS:
            ;2456 ;
            ;2457 ; d - Contains aligned starting address (B Register Image)
            ;2458 ;
            ;2459 ; SIDE EFFECTS:
            ;2460 ;
            ;2461 ; d,q,va, and sc are destroyed
            ;2462 ;--
            ;2463 SET.START.ADDR:
U 11AD, 0010,0038,6580,0970,0284,71AE  ;2464  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11AE, 0000,003C,01F8,0803,0000,11AF  ;2465  va_va+4,q_0  ; Set address to Register B
            ;2466  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11AF, 0D00,003C,0980,0800,0084,B1B0  ;2467      sc_sc-k[.2]  ; Setup to insert bit 14
U 11B0, 0801,001C,0180,0800,0000,11B1  ;2468  d_d.ornot.mask  ; Insert Write Enable bit
U 11B1, 0000,003E,0180,A800,0000,0001  ;2469  cache.p_d[long],return1 ; Set the starting address and return
            ;2470
```

```
;2471   .PAGE
;2472   .TOC "   CONFIGURE MS780-E/H"
;2473 ;*************************************************************
;2474 ;++
;2475 ;
;2476 ; Functional Description:
;2477 ; ----------------------
;2478 ;
;2479 ;      This  subroutine  will  be  used  by  tests  that  do  not
;2480 ; specifically  check  the  memory, but  make use of it to execute.
;2481 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2482 ; that  if  a  Read/Write operation will take place no false errors
;2483 ; will be logged due to misconfigured memory.
;2484 ;
;2485 ; Calling Constraint: =00
;2486 ; ------------------
;2487 ;
;2488 ;
;2489 ; Inputs:
;2490 ; ------
;2491 ;
;2492 ; RC[0E] - I/O BASE OF MS780-E/H
;2493 ;
;2494 ;
;2495 ; Outputs:
;2496 ; -------
;2497 ;
;2498 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2499 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2500 ;      OR IF NO MS780-E/Hs WERE FOUND
;2501 ;      ON THE SBI
;2502 ;
;2503 ; Side Effects:
;2504 ; ------------
;2505 ;
;2506 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2507 ;
;2508 ;
;2509 ;
;2510 ;--
;2511 ;*************************************************************
;2512 CONFIG.MS780E:
U 11B2, 0818,0038,0D80,0800,0000,11B3  ;2513  D_K[.3]    ; Set up flag for the Fail Chain
U 11B3, 0001,003C,0180,09E8,0000,1004  ;2514  RC[0D]_D    ; ...
U 1004, 0000,003D,0180,0800,0000,1042  ;2515 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1005, 0000,003D,0180,0800,0000,112C  ;2516  ERROR1     ; No MS780-E/H's found on the SBI
U 1006, 0000,003C,0180,0800,0000,1010  ;2517  NOP     ; OK. Found at least one MS780-E/H
;2518 =
;2519 CONFIG.RETRY:     ; Entry point for the case in which the
;2520       ; ...first MS780-E/H could not be
;2521       ; ...properly configured
;2522 =00 D_K[ZERO],    ;
U 1010, 0818,0039,1980,0800,0000,11B5  ;2523   CALL[SELECT.CNTRLR] ; Try to select the lower controller
U 1011, 0000,003C,0180,0800,0000,1014  ;2524  J/TRY.UPPER    ; Could NOT configure the lower
;2525       ; ...controller, try upper
```

```
U 1012, 0000,003E,0180,0800,0000,0001  ;2526  RETURN1    ; OK. Lower Controller is configured
            ;2527 =
            ;2528 =00
            ;2529 TRY.UPPER:
U 1014, 0818,0039,0980,0800,0000,11B5  ;2530  D_K[.2],CALL[SELECT.CNTRLR] ; Try to select upper controller
U 1015, 0000,003C,0180,0800,0000,1018  ;2531  J/CNFG.TRY.NEXT   ; Could not config. either controller
U 1016, 0000,003E,0180,0800,0000,0001  ;2532  RETURN1    ; OK. Upper controller configured
            ;2533 =
            ;2534 =00
            ;2535 CNFG.TRY.NEXT:
U 1018, 0000,003D,0180,0800,0000,11BF  ;2536  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2537       ; ...MS780-E/H found so go and see
            ;2538       ; ...if there are any more
U 1019, 0000,003C,0180,0800,0000,11B4  ;2539  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2540       ; ...try to configure it
U 101A, 0818,0038,0D80,0800,0000,101B  ;2541  D_K[.3]    ; Set Flag for Fail Chain
U 101B, 0001,003C,0180,09E8,0000,11B4  ;2542  RC[0D]_D    ; ...
U 11B4, 0000,003D,0180,0800,0000,112C  ;2543  ERROR1    ; Could not configure any MS780-E/H
            ;2544       ; ...abort the test
            ;2545 =
            ;2546
```

```
            ;2547  .PAGE
            ;2548  .TOC "   Select Controller"
            ;2549 ;**
            ;2550 ; FUNCTIONAL DESCRIPTION:
            ;2551 ;
            ;2552 ; This routine is used to put the memory system into the
            ;2553 ; specified configuration with respect to controller select.
            ;2554 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2555 ; a RETURN2 is made.
            ;2556 ;
            ;2557 ; CALLING CONSTRAINT:
            ;2558 ;
            ;2559 ; =00
            ;2560 ;
            ;2561 ; INPUTS:
            ;2562 ;
            ;2563 ; d - Contains value to write to bits<2:0> of Register A
            ;2564 ; rc[0e] - Address of Register A
            ;2565 ;
            ;2566 ; SIDE EFFECTS:
            ;2567 ;
            ;2568 ; sc,d,va are destroyed
            ;2569 ;--
            ;2570 SELECT.CNTRLR:
U 11B5, 0010,0038,0180,0970,0284,71B6  ;2571  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11B6, 0801,001C,0180,0800,0000,11B7  ;2572  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11B7, 0000,003C,0180,A800,0000,11B8  ;2573  cache.p_d[long]  ; Write the configuration Register
U 11B8, 0818,0038,5D80,0800,0000,11B9  ;2574  d_k[.7]   ; Get Misconfiguration bits
U 11B9, 0000,003C,61F8,0800,0084,71BA  ;2575  sc_k[.F],q_0  ; ...
U 11BA, 0D00,003C,0180,0800,0000,11BB  ;2576  d_dal.sc  ; ... D = x38000
U 11BB, 0000,003C,01E0,0800,0000,11BC  ;2577  q_d   ; Save mask
U 11BC, 0000,003C,0180,C800,0000,11BD  ;2578  d[long]_cache.p  ; Read CNFG A Reg and
            ;2579  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11BD, 001D,2034,0180,0800,0010,11BE  ;2580   clk.ubcc ; ...
U 11BE, 0000,013C,0180,0800,0000,1060  ;2581  z?   ; Branch if no
U 1060, 0000,003E,0180,0800,0000,0001  ;2582 =0 RETURN1   ; Bad configuration
U 1061, 0000,003E,0180,0800,0000,0002  ;2583  RETURN2   ; Configuration ok
            ;2584
            ;2585
```

```
            ;2586  .PAGE
            ;2587  .TOC "   ENABLE NEXT MS780-E"
            ;2588 ;**
            ;2589 ; FUNCTIONAL DESCRIPTION:
            ;2590 ;
            ;2591 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2592 ;    They will be tested  separately, according to the TR level at which
            ;2593 ;    they are located,  starting with the one  at the lowest level first.
            ;2594 ;          When  a test has finished  with the first unit, it will have to call
            ;2595 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2596 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2597 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2598 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2599 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2600 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2601 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2602 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2603 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2604 ;
            ;2605 ; CALLING CONSTRAINT:
            ;2606 ;
            ;2607 ;  =00
            ;2608 ;
            ;2609 ;--
            ;2610 ENABLE.NEXT.MS780E:
U 11BF, 0000,003C,F9F0,2C00,0000,11C0  ;2611  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2612  ALU_Q,    ; Check to see if it is zero
U 11C0, 0001,203C,0180,0800,0010,11C1  ;2613      CLK.UBCC  ; Check Condition Codes
U 11C1, 0000,013C,0180,0800,0000,1062  ;2614  Z?    ; ...for zero
U 1062, 0000,003C,0180,0800,0000,11C2  ;2615 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1063, 0000,003E,0180,0800,0000,0002  ;2616  RETURN2    ; Zero No more units to test
            ;2617 ENABLE.NEXT:
U 11C2, 0818,0038,C180,0800,0000,1064  ;2618  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1064, 0000,003D,0180,0800,0000,11AD  ;2619 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1065, 0818,0038,7980,0800,0000,11C3  ;2620  D_K[.30]  ; Prepare to point to the ID register holding
            ;2621      ; ...the I/O Base address of the next MS780-E
U 11C3, 0819,0014,1180,0800,0000,11C4  ;2622  D_D+K[.4]  ; ...
U 11C4, 0000,003C,F580,0800,0000,11C5  ;2623  K[.A]    ; ...
U 11C5, 0819,0014,F580,0800,0000,11C6  ;2624  D_D+K[.A]  ; ...
U 11C6, 0000,003C,F9F0,2C00,0000,11C7  ;2625  Q_ID[3E]  ; Get MS780-E Counter
            ;2626  D_D-Q,    ; D now holds the pointer to the ID register
U 11C7, 081D,0000,0180,0800,0082,11C8  ;2627    SC_ALU  ; ...
U 11C8, 0000,003C,01F0,2400,0000,11C9  ;2628  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11C9, 0001,203C,0180,09F0,0000,11CA  ;2629  RC[0E]_Q  ; Save it also in RC[0E]
U 11CA, 0F03,003C,0180,09E8,0000,1066  ;2630  RC[0D]_0,D_0  ; Set starting address to zero
U 1066, 0000,003D,0180,0800,0000,11AD  ;2631 =0 CALL[SET.START.ADDR] ; ....
U 1067, 0F00,003C,F9F0,2C00,0000,11CB  ;2632  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11CB, 081D,2008,0180,0800,0000,11CC  ;2633  D_Q-D-1    ; Subtract 1 from total
U 11CC, 0000,003C,F980,3C00,0000,11CD  ;2634  ID[3E]_D  ; Adjust the pointer
U 11CD, 0000,003E,0180,0800,0000,0001  ;2635  RETURN1    ; Tell caller another unit was found
            ;2636
            ;2637
            ;2638
            ;2639 ;*
            ;2640 ; THE FOLLOWING ROUTINE WAITS FOR 4 CYCLES FOR THE IB TO GET A DATA WORD
```

```
            ;2641 ; THIS ROUTINE MUST BE CALLED WITHIN 3 OR 4 INSTRUCTIONS OF THE "IBC/START"
            ;2642 ; FUNCTION. IF LESS THAN 3, THE IB WILL NOT GET ANY DATA. IF MORE THAN
            ;2643 ; 4, THE IB WILL GET CACHE DATA ON THE SECOND REFERANCE.
            ;2644 ;-
            ;2645
U 11CE. 0000.003C.0180.F800.0000.11CF  ;2646 WAIT7: MCT/ALLOW.IB.READ
U 11CF. 0000.003C.0180.F800.0000.11D0  ;2647  MCT/ALLOW.IB.READ
U 11D0. 0000.003C.0180.F800.0000.11D1  ;2648  MCT/ALLOW.IB.READ
U 11D1. 0000.003C.0180.F800.0000.11D2  ;2649  MCT/ALLOW.IB.READ
U 11D2. 1000.003E.0180.F800.0000.0001  ;2650  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2651
            ;2652
            ;2653 ;*
            ;2654 ; THE FOLLOWING ROUTINE WAITS FOR AN IB REFERANCE TO TIMEOUT
            ;2655 ;-
            ;2656
            ;2657 WAITFORTIMEOUT:
U 11D3. 0018.0038.49C0.F800.0000.11D4  ;2658  MCT/ALLOW.IB.READ,Q_K[.FF] ; SETUP A COUNTER
U 11D4. 0000.003C.03A8.F800.0000.11D5  ;2659  MCT/ALLOW.IB.READ,Q_Q.LEFT,SI/MUL- ; ...
U 11D5. 0019.2000.05C0.F800.0010.11D6  ;2660 TOWAIT: MCT/ALLOW.IB.READ,Q_Q-K[.1],CLK.UBCC ; CHECK THE COUNTER
U 11D6. 0000.013C.0180.F800.0000.1068  ;2661  MCT/ALLOW.IB.READ,Z?  ; DONE YET?
U 1068. 0000.003C.0180.F800.0000.11D5  ;2662 =0 MCT/ALLOW.IB.READ,J/TOWAIT ; NO
U 1069. 0000.003E.0180.0800.0000.0001  ;2663  RETURN1    ; EXIT
            ;2664
            ;2665 ;*
            ;2666 ; THIS ROUTINE SETSUP AND EXECUTES AN INSTRUCTION BUFFER AUTO REFILL
            ;2667 ; HIT. IT EXITS IN THE CYCLE THAT STARTS THE AUTO REFILL.
            ;2668 ;-
            ;2669
            ;2670 AUTOREFILLHIT:
U 11D7. 2000.003C.0180.0A28.4200.11D8  ;2671  ALU_R[5],FLUSH.IB  ; LOAD THE VIBA
U 11D8. 0000.003C.0180.6000.0000.11D9  ;2672  LOAD.IB    ; LOAD THE IPA
U 11D9. 3000.003C.0180.0A20.0200.11DA  ;2673  ALU_R[4],VA_ALU,IBC/START ; RELOAD THE VA
U 11DA. 0000.003C.0180.F800.0000.11DB  ;2674  MCT/ALLOW.IB.READ  ; ALLOW THE IPA TO COUNT
U 11DB. 0000.003E.0180.F800.0000.0001  ;2675  MCT/ALLOW.IB.READ,RETURN1 ; THIS CYCLE STARTS THE AUTO REFILL
            ;2676
            ;2677
            ;2678 ;*
            ;2679 ; THIS ROUTINE SETSUP AND EXECUTES AN INSTRUCTION BUFFER AUTO REFILL MISS.
            ;2680 ; IT EXITS IN THE CYCLE THAT STARTS THE AUTO REFILL.
            ;2681 ;-
            ;2682
            ;2683 AUTOREFILLMISS:
U 11DC. 2000.003C.0180.0A28.4200.11DD  ;2684  ALU_R[5],FLUSH.IB  ; LOAD THE VIBA
U 11DD. 0000.003C.0180.6000.0000.11DE  ;2685  LOAD.IB    ; LOAD THE IPA
U 11DE. 3000.003C.0180.0A20.0200.11DF  ;2686  ALU_R[4],VA_ALU,IBC/START ; RELOAD THE VA
U 11DF. 0000.003C.0180.F800.0000.11E0  ;2687  MCT/ALLOW.IB.READ  ; WAIT
U 11E0. 0000.003C.0180.F800.0000.11E1  ;2688  MCT/ALLOW.IB.READ  ; WAIT
U 11E1. 0000.003C.0180.F800.0000.11E2  ;2689  MCT/ALLOW.IB.READ  ; WAIT
U 11E2. 0000.003C.0180.F800.0000.11E3  ;2690  MCT/ALLOW.IB.READ  ; WAIT
U 11E3. 0000.003C.0180.F800.0000.11E4  ;2691  MCT/ALLOW.IB.READ  ; WAIT
U 11E4. 0000.003C.0180.F800.0000.11E5  ;2692  MCT/ALLOW.IB.READ  ; WAIT
U 11E5. 0000.003C.0180.F800.0000.11E6  ;2693  MCT/ALLOW.IB.READ  ; WAIT
U 11E6. 0000.003C.0180.F800.0000.11E7  ;2694  MCT/ALLOW.IB.READ  ; WAIT
U 11E7. 0000.003C.0180.F800.0000.11E8  ;2695  MCT/ALLOW.IB.READ
```

```
U 11E8, 0000,003E,0180,F800,0000,0001  ;2696  MCT/ALLOW.IB.READ,RETURN1 ; WAIT
        ;2697
        ;2698
        ;2699 ;*
        ;2700 ; THIS ROUTINE LOOKS FOR A DEVICE ON THE SBI AND RETURNS WITH THE DEVICES
        ;2701 ; BASE ADDRESS IN R[0] IF IT FINDS ONE, OTHERWISE R[0] IS CLEAR.
        ;2702 ;-
        ;2703
        ;2704 LOOKFORDEVICE:
U 11E9, 0818,0038,7580,0800,0000,11EA  ;2705  D_K[.20]
U 11EA, 0B00,003C,0180,0800,0000,11EB  ;2706  D_D.SWAP    ; GENERATE HIGH WORD OF DEVICE ADDRESS
U 11EB, 0001,003C,0180,0A80,0000,11EC  ;2707  R[0]_D     ; SAVE
U 11EC, 0818,0038,75F8,0800,0000,11ED  ;2708  D_K[.20],Q_0    ; GENERATE INITIAL LOW WORD
U 11ED, 0000,003C,0180,0800,0084,71EE  ;2709  SC_K[.8]    ; ...
U 11EE, 0D00,003C,0180,0800,0000,11EF  ;2710  D_DAL.SC    ; ...
U 11EF, 081C,2030,0180,0A00,0000,11F0  ;2711  D_D.OR.R[0]    ; D= 20002000
U 11F0, 0001,003C,0180,0A80,0000,11F1  ;2712  R[0]_D     ; SAVE (ADRS OF DEVICE AT TR=1)
U 11F1, 0818,0038,7580,0800,0000,11F2  ;2713  D_K[.20]    ; GENERATE INCREMENT FOR DEVICE ADDRESS
U 11F2, 0D00,003C,0180,0800,0000,11F3  ;2714  D_DAL.SC    ; D= 2000
U 11F3, 0000,003C,01E0,0800,0000,11F4  ;2715  Q_D     ; SAVE INCREMENT IN Q
U 11F4, 0000,003C,0180,0A00,0200,11F5  ;2716  VA_ALU,ALU_R[0]    ; LOAD THE DEVICE ADDRESS
U 11F5, 0018,0038,6180,0A90,0000,11F6  ;2717  R[2]_K[.F]    ; SET THE LOOP COUNT
U 11F6, 0000,003C,8980,0800,0084,71F7  ;2718  SC_K[.D]
        ;2719 TRYAGAIN:
U 11F7, 0000,003C,0180,0A00,0200,11F8  ;2720  VA_ALU,ALU_R[0]    ; LOAD THE ADDRESS
U 11F8, 0003,001C,0180,0AF8,0000,11F9  ;2721  R[0F]_NOT.MASK    ; SET THE TIMEOUT TRAP EXPECTED FLAG
U 11F9, 0000,003C,0180,C800,0000,11FA  ;2722  D[LONG]_CACHE.P    ; READ THE ADDRESS
U 11FA, 0000,003C,0180,0A78,0010,11FB  ;2723  ALU_R[0F],CLK.UBCC  ; HAS THERE A TIMEOUT?
U 11FB, 0000,013C,0180,0800,0000,106A  ;2724  Z?     ; BRANCH IF YES
U 106A, 0000,003C,0180,0800,0000,1208  ;2725 =0 J/FOUNDONE    ; EXIT
        ;2726 FOUNDMEMORY:
U 106B, 0818,0038,4580,0800,0000,11FC  ;2727  D_K[.8000]    ; CLEAR THE TIMEOUT BIT
U 11FC, 0200,003C,0180,0800,0000,11FD  ;2728  D_D.RIGHT2
U 11FD, 0600,003C,0180,0800,0000,11FE  ;2729  D_D.RIGHT
U 11FE, 0000,003C,6580,3C00,0000,11FF  ;2730  ID[SBI.ERR]_D
U 11FF, 0800,003C,0180,0A00,0000,1200  ;2731  D_R[0]     ; GET THE CURRENT ADDRESS
U 1200, 001D,0014,0180,0A80,0200,1201  ;2732  ALU_D+Q,VA_ALU,R[0]_ALU  ; INCREMENT THE ADDRESS
U 1201, 0800,003C,0180,0A10,0000,1202  ;2733  D_R[2]     ; GET THE LOOP COUNT
U 1202, 0019,0000,0580,0A90,0010,1203  ;2734  R[2]_D-K[.1],CLK.UBCC  ; DECREMENT AND CHECK
U 1203, 0000,013C,0180,0800,0000,106C  ;2735  Z?     ; BRANCH IF DONE
U 106C, 0000,003C,0180,0800,0000,11F7  ;2736 =0 J/TRYAGAIN    ; CONTINUE LOOKING FOR DEVICE
U 106D, 0003,003C,0180,0A80,0084,7204  ;2737  R[0]_0,SC_K[.8]    ; NO DEVICE AVAILABLE
U 1204, 0818,0038,65F8,0800,0000,1205  ;2738  D_K[.10],Q_0    ; SETUP TO CLEAR SBI ERROR REG
U 1205, 0D00,003C,0180,0800,0000,1206  ;2739  D_DAL.SC
U 1206, 0819,0030,3180,0800,0000,1207  ;2740  D_D.OR.K[.40]
U 1207, 0000,003E,6580,3C00,0000,0001  ;2741  ID[SBI.ERR]_D,RETURN1  ; CLEAR ERROR REG AND RETURN
        ;2742 FOUNDONE:
U 1208, 0200,003C,0180,0800,0000,1209  ;2743  D_D.RIGHT2    ; CHECK BIT 5 TO SEE IF ITS MEMORY
U 1209, 0200,003C,0180,0800,0000,120A  ;2744  D_D.RIGHT2    ; ...
U 120A, 0600,003C,0180,0800,0000,120B  ;2745  D_D.RIGHT    ; ...
U 120B, 0819,0024,5180,0800,0000,120C  ;2746  D_D.ANDNOT.K[.1E]  ; ...
U 120C, 0000,193C,0180,0800,0000,101C  ;2747  D3-0?
U 101C, 0000,003C,0180,0800,0000,106B  ;2748 =1100 J/FOUNDMEMORY    ; MS780-A/C
U 101D, 0000,003C,0180,0800,0000,120D  ;2749  J/FOUND.UBA.MBA    ; Could be UBA or MBA
U 101E, 0000,003C,0180,0800,0000,106B  ;2750  J/FOUNDMEMORY    ; Unidentified device. Ignore it
```

```
U 101F. 0000.003C.0180.0800.0000.106B  ;2751  J/FOUNDMEMORY   ; MS780-E/H memory
          ;2752 FOUND.UBA.MBA:
U 120D. 0000.003C.0180.0800.0084.720E  ;2753  SC_K[.8]
U 120E. 0818.0038.6580.0800.0000.120F  ;2754  D_K[.10]   ; SETUP TO CLEAR ERROR REG
U 120F. 0D00.003C.01F8.0800.0000.1210  ;2755  D_DAL.SC,Q_0
U 1210. 0819.0030.3180.0800.0000.1211  ;2756  D_D.OR.K[.40]
U 1211. 0000.003E.6580.3C00.0000.0001  ;2757  ID[SBI.ERR]_D,RETURN1  ; CLEAR ERROR REG AND RETURN
          ;2758
```

```
                ;2759  .PAGE
                ;2760  .TOC "   WAIT REFRESH SUBROUTINE"
                ;2761 ;*****************************************************************
                ;2762 ;++
                ;2763 ;
                ;2764 ; Functional Description:
                ;2765 ; ---------------------
                ;2766 ;
                ;2767 ; This subroutine provides the means of synchronizing the
                ;2768 ; diagnostic with the end of a refresh cycle on a MS780-E/H
                ;2769 ; memory.  It should be used by all the tests in which
                ;2770 ; timing is critical and delays due to refresh could
                ;2771 ; introduce false errors.
                ;2772 ;
                ;2773 ; Calling Constraint: =0
                ;2774 ; ------------------
                ;2775 ;
                ;2776 ;
                ;2777 ; Inputs:
                ;2778 ; ------
                ;2779 ;
                ;2780 ; R[0E] must hold the CNFG Reg B Address
                ;2781 ;
                ;2782 ; NOTE: This is slightly different than the similar subroutine
                ;2783 ;  used in ESKAR42 and ESKAR43.  This change is necessary
                ;2784 ;  due to different register assignments in these sections.
                ;2785 ;
                ;2786 ;
                ;2787 ; Outputs:
                ;2788 ; -------
                ;2789 ;
                ;2790 ; RETURN1 at the end of a refresh cycle.
                ;2791 ;
                ;2792 ;
                ;2793 ; Side Effects:
                ;2794 ; ------------
                ;2795 ;
                ;2796 ; The contents of the following registers will be lost:
                ;2797 ;
                ;2798 ; D, SC, CNFG Reg B
                ;2799 ;
                ;2800 ;--
                ;2801 ;*****************************************************************
                ;2802 WAITREFRESH:
U 1212, 0000,003C,0180,0A70,0200,1213  ;2803  VA_R[0E]  ; Get address of CNFG Reg B
U 1213, 0000,003C,0180,0800,0084,7214  ;2804  SC_K[.B]  ; Get data for CNFG  Reg B
U 1214, 0803,001C,0180,0800,0000,1215  ;2805  D_NOT.MASK  ; Set bit (Refresh Lock Bit)
U 1215, 0000,003C,0180,A800,0000,1216  ;2806  CACHE.P_D[LONG]  ; Set bit 8 in CNFG Reg B
U 1216, 0F00,003C,0180,0800,0000,1217  ;2807  D_0  ; Get ready to clear the bit
U 1217, 0000,003C,0180,A800,0000,1218  ;2808  CACHE.P_D[LONG]  ; Clear bit 8 in CNFG Reg B
                ;2809 REFRESH.WAIT.LOOP:
                ;2810  D[LONG]_CACHE.P, ; Get contents of CNFG Reg B
U 1218, 0000,003C,0180,C800,0084,7219  ;2811  SC_K[.8] ; Point to bit 8
                ;2812  ALU_D.ANDNOT.MASK, ; Is bit 8 still set?
U 1219, 0001,0024,0180,0800,0010,121A  ;2813  CLK.UBCC ; ...
```

```
U 121A. 0000,013C,0180,0800,0000,106E  ;2814  Z?   ;
U 106E. 0000,003C,0180,0800,0000,1218  ;2815 =0 J/REFRESH.WAIT.LOOP ; Bit 8 is still set
U 106F. 0818,0038,1180,0800,0000,1070  ;2816  D_K[.4]   ; Number of cycles to wait for
          ;2817     ; ...refersh to end
U 1070. 0000,003D,0180,0800,0000,121B  ;2818 =0 CALL[WAIT.LOOP]  ; Go and wait for refresh to end
U 1071. 0000,003E,0180,0800,0000,0001  ;2819  RETURN1   ; OK. Refresh should be done by now
          ;2820     ; ...Return to caller
          ;2821
```

```
            ;2822  .PAGE
            ;2823  .TOC "   Wait Loop Routine"
            ;2824 ;++
            ;2825 ; FUNCTIONAL DESCRIPTION:
            ;2826 ;
            ;2827 ; This routine is used to wait the specified number of machine cycles.
            ;2828 ; This routine is typically called to wait for an SBI write to
            ;2829 ; I/O space to complete.
            ;2830 ;
            ;2831 ; CALLING CONSTRAINT:
            ;2832 ;
            ;2833 ; =0
            ;2834 ;
            ;2835 ; INPUTS:
            ;2836 ;
            ;2837 ; d - Contains number of cycles to wait
            ;2838 ; alu.z condition code must not be set
            ;2839 ;
            ;2840 ; SIDE EFFECTS:
            ;2841 ;
            ;2842 ; d is destroyed
            ;2843 ;--
            ;2844 WAIT.LOOP:
U 121B, 0018,0038,6180,0800,0010,1072  ;2845  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2846      ; ...is not set
            ;2847 =0
            ;2848 W.LOOP:
            ;2849  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1072, 0819,0100,0580,0800,0010,1072  ;2850       j/W.LOOP
U 1073, 0000,003E,0180,0800,0000,0001  ;2851  return1    ; exit
            ;2852
            ;2853
```

```
                ;2854 .PAGE "TEST 1E9 AUTO REFILL HIT AND CP READ HIT"
                ;2855 ;*****************************************************************
                ;2856 ;++
                ;2857 ;
                ;2858 ; AUTO REFILL HIT AND CP READ HIT
                ;2859 ;
                ;2860 ; TEST DESCRIPTION
                ;2861 ;
                ;2862 ; THIS TEST EXECUTES AN AUTO REFILL THAT IS A HIT AND A CP READ
                ;2863 ; THAT IS ALSO A HIT IN THE SAME CYCLE. A CHECK IS MADE THAT A STALL
                ;2864 ; OCCURRED FOR ONE CYCLE AND THAT THE CP RECEIVED THE CORRECT DATA.
                ;2865 ; THE CP DATA IS ALL ZERO'S AND THE IB DATA IS ALL ONE'S.
                ;2866 ;
                ;2867 ; BIT 8 OF THE TB ERROR REGISTER 0 IS ALSO TESTED.
                ;2868 ;
                ;2869 ; LOGIC DESCRIPTION
                ;2870 ;
                ;2871 ; THIS TEST CHECKS PART OF THE CANCEL SIGNAL ON THE TBM MODULE
                ;2872 ; AND PART OF THE STALL LOGIC ON THE SBL MODULE.
                ;2873 ; IT ALSO CHECKS BIT 8 OF THE TB ERROR REGISTER 0 ON THE TBM MODULE.
                ;2874 ;
                ;2875 ; ERROR DESCRIPTION
                ;2876 ;
                ;2877 ; THE ERROR TYPEOUT IS DEFINED AS FOLLOWS:
                ;2878 ;
                ;2879 ; ERR0: DATA EXPECTED SILO COUNTER
                ;2880 ;   RECEIVED SILO COUNTER
                ;2881 ; ERR1: DATA EXPECTED CPU DATA
                ;2882 ;   RECEIVED CPU DATA
                ;2883 ; ERR1A: DATA EXPECTED CONTENTS OF TB ERR REG 0
                ;2884 ;   RECEIVED CONTENTS OF TB ERR REG 0
                ;2885 ; ERR1B: SAME AS ERR1A
                ;2886 ;--
                ;2887 ;*****************************************************************
                ;2888 =0
U 1074, 0018,0039,0980,09F8,0000,1116  ;2889 T001: NEWTST[.2]
U 1075, 0000,003C,0180,0800,0000,1076  ;2890  NOP
U 1076, 0000,003D,0180,0800,0000,1036  ;2891 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 1077, 0F00,003C,0180,0A20,0200,121C  ;2892  ALU_R[4],VA_ALU,D_0  ; LOAD THE CPU ADDRESS
U 121C, 0000,003C,0180,9800,0000,121D  ;2893  MCT/VALIDATE   ; MAKE IT A HIT
U 121D, 0000,003C,0180,0803,0000,121E  ;2894  VA_VA+4     ; AND THE NEXT LONG WORD
U 121E, 0000,003C,0180,9800,0000,121F  ;2895  MCT/VALIDATE
U 121F, 0000,003C,0180,0A28,0200,1220  ;2896  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 1220, 0818,0038,C180,0800,0000,1221  ;2897  D_K[.FFFF]   ; GENERATE DATA PATTERN TO WRITE
U 1221, 0802,403C,0180,0800,0000,1222  ;2898  D_D.SXT[WORD]   ; ...
U 1222, 0000,003C,0180,9800,0000,1223  ;2899  MCT/VALIDATE   ; WRITE THE DATA PATTERN
U 1223, 0000,003C,0180,0803,0000,1224  ;2900  VA_VA+4
U 1224, 0000,003C,0180,9800,0000,1078  ;2901  MCT/VALIDATE   ; ...
U 1078, 0000,003D,0180,0800,0000,112B  ;2902 =0 ERLOOP
U 1079, 0818,0038,5DF8,0800,0000,1225  ;2903  D_K[.7],Q_0   ; GENERATE EXPECTED SILO COUNTER
U 1225, 0000,003C,6580,0800,0084,7226  ;2904  SC_K[.10]
U 1226, 0D00,003C,0180,0800,0000,1227  ;2905  D_DAL.SC
U 1227, 0001,003C,0180,09F0,0000,1228  ;2906  RC[0E]_D   ; SAVE
U 1228, 0818,0038,7580,0800,0000,1229  ;2907  D_K[.20]   ; GENERATE PATTERN TO TURN SILO
U 1229, 0B00,003C,0180,0800,0000,107A  ;2908  D_D.SWAP   ; COUNTER ON
```

```
U 107A, 0000,003D,7180,3C00,0000,11D7  ;2909 =0 ID[COMP]_D,CALL,J/AUTOREFILLHIT ; GO EXECUTE THE AUTO REFILL
U 107B, 0000,003C,0180,C800,0000,122A  ;2910  D[LONG]_CACHE.P   ; NOW THE CP READ AT THE SAME TIME
U 122A, 1001,003C,71F0,2E88,0000,122B  ;2911  Q_ID[COMP],IBC/STOP,R[1]_D ; READ THE SILO COUNTER
U 122B, 001C,0034,01C0,0A60,0000,122C  ;2912  Q_Q.AND.R[0C]   ; MASK
U 122C, 0810,0038,0180,0970,0000,122D  ;2913  D_RC[0E]   ; GET THE EXPECTED DATA
U 122D, 001D,2000,0180,0800,0010,122E  ;2914  ALU_Q-D,CLK.UBCC  ; CHECK
U 122E, 0000,013C,0180,0800,0000,107C  ;2915  Z?    ; BRANCH IF OK
          ;2916 =0
U 107C, 0001,203D,0180,09E8,0000,112E  ;2917 ERR0: ERROR2,RC[0D]_Q   ; STALL DID NOT OCCUR
          ;2918
          ;2919 ;*
          ;2920 ; NOW CHECK THAT THE CPU RECEIVED THE CORRECT DATA
          ;2921 ;-
          ;2922
U 107D, 0F03,003C,0180,09F0,0000,122F  ;2923  RC[0E]_0,D_0   ; SET THE EXPECTED DATA
U 122F, 0000,003C,01C0,0A08,0000,1230  ;2924  Q_R[1]    ; GET THE RECEIVED DATA
U 1230, 001D,2000,0180,0800,0010,1231  ;2925  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 1231, 0000,013C,0180,0800,0000,107E  ;2926  Z?    ; BRANCH IF OK
          ;2927 =0
U 107E, 0001,203D,0180,09E8,0000,112E  ;2928 ERR1: ERROR2,RC[0D]_Q   ; CPU RECEIVED THE WRONG DATA
          ;2929
U 107F, 0000,003C,0180,0800,0000,1232  ;2930  NOP
          ;2931
          ;2932 ;*
          ;2933 ; NOW CHECK THAT BIT 8 IN TB ERROR REG 0 SET
          ;2934 ;-
U 1232, 0818,0038,4180,0800,0000,1233  ;2935  D_K[.80]   ; GENERATE THE EXPECTED DATA
U 1233, 0500,003C,0180,0800,0000,1234  ;2936  D_D.LEFT
U 1234, 0001,003C,49F0,2DF0,0000,1235  ;2937  RC[0E]_D,Q_ID[TBER0]  ; READ THE ERROR REGISTER
U 1235, 001D,0034,01C0,0800,0000,1236  ;2938  Q_ALU,ALU_D.AND.Q  ; MASK BIT 8
U 1236, 001D,2000,0180,0800,0010,1237  ;2939  ALU_Q-D,CLK.UBCC  ; CHECK THAT BIT 8 SET
U 1237, 0000,013C,0180,0800,0000,1080  ;2940  Z?
          ;2941 =0
U 1080, 0001,203D,0180,09E8,0000,112E  ;2942 ERR1A: ERROR2,RC[0D]_Q   ; TBER0 BIT 8 DID NOT SET
          ;2943
          ;2944 ;*
          ;2945 ; NOW CHECK THAT THE BIT CLEARS WHEN AUTO RELOAD IS FALSE
          ;2946 ;-
          ;2947
U 1081, 0810,0038,0180,0970,0000,1238  ;2948  D_RC[0E]   ; GET THE MASK FOR BIT 8
U 1238, 0003,003C,0180,09F0,0000,1239  ;2949  RC[0E]_0   ; SET THE EXPECTED DATA
U 1239, 0000,003C,0180,F800,0000,123A  ;2950  MCT/ALLOW.IB.READ  ; BIT SHOULD CLEAR
U 123A, 0000,003C,49F0,2C00,0000,123B  ;2951  Q_ID[TBER0]   ; READ THE ERROR REG
U 123B, 001D,0034,01C0,0800,0010,123C  ;2952  Q_ALU,ALU_D.AND.Q,CLK.UBCC ; MASK AND CHECK FOR ZERO
U 123C, 0000,013C,0180,0800,0000,1082  ;2953  Z?
          ;2954 =0
U 1082, 0001,203D,0180,09E8,0000,112E  ;2955 ERR1B: ERROR2,RC[0D]_Q   ; TBER0 BIT 8 DID NOT CLEAR
U 1083, 0000,003C,0180,0800,0000,1084  ;2956 END001: NOP
          ;2957
          ;2958
```

```
          ;2959 .PAGE "TEST 1EA AUTO REFILL MISS AND CP READ MISS"
          ;2960 ;*****************************************************************
          ;2961 ;**
          ;2962 ;
          ;2963 ; AUTO REFILL MISS AND CP READ MISS
          ;2964 ;
          ;2965 ; TEST DESCRIPTION
          ;2966 ;
          ;2967 ; THIS TEST CHECKS THAT THE CPU MEMORY CYCLE IS MADE TO THE CORRECT
          ;2968 ; ADDRESS WHEN IT IS INITIATED DURING AN AUTO REFILL.
          ;2969 ;
          ;2970 ; LOGIC DESCRIPTION
          ;2971 ;
          ;2972 ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT INHIBITS THE START OF THE
          ;2973 ; BUS CYCLE UNTIL THE AUTO REFILL IS COMPLETE.
          ;2974 ;
          ;2975 ; ERROR DESCRIPTION
          ;2976 ;
          ;2977 ; DATA: EXPECTED CPU DATA
          ;2978 ;       RECEIVED CPU DATA
          ;2979 ;--
          ;2980 ;*****************************************************************
          ;2981 =0
U 1084, 0018,0039,0980,09F8,0000,1116  ;2982 T002: NEWTST[.2]
U 1085, 0000,003C,0180,0800,0000,1086  ;2983  NOP
U 1086, 0000,003D,0180,0800,0000,1036  ;2984 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 1087, 0003,003C,0180,09F0,0000,123D  ;2985  RC[0E]_0   ; SET THE EXPECTED CPU DATA
U 123D, 0F00,003C,0180,0A20,0200,123E  ;2986  ALU_R[4],VA_ALU,D_0  ; SETUP TO WRITE CPU DATA
U 123E, 0000,003C,0180,A800,0000,123F  ;2987  CACHE.P_D[LONG]   ; WRITE THE CPU DATA
U 123F, 0818,0038,C180,0800,0000,1240  ;2988  D_K[.FFFF]   ; GENERATE IB DATA
U 1240, 0802,403C,0180,0800,0000,1241  ;2989  D_D.SXT[WORD]   ; ...
U 1241, 0000,003C,0180,0A28,0200,1242  ;2990  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 1242, 0000,003C,0180,A800,0000,1243  ;2991  CACHE.P_D[LONG]   ; WRITE IT
U 1243, 0000,003C,0180,0803,0000,1244  ;2992  VA_VA+4
U 1244, 0000,003C,0180,A800,0000,1088  ;2993  CACHE.P_D[LONG]   ; AND THE NEXT LONG WORD
U 1088, 0000,003D,0180,0800,0000,112B  ;2994 =0 ERLOOP
U 1089, 0000,003C,0180,0A20,0200,1245  ;2995  ALU_R[4],VA_ALU   ; LOAD THE CPU ADDRESS
U 1245, 0000,003C,0180,9000,0000,1246  ;2996  CACHE.INVALIDATE  ; MAKE SHURE ITS NOT A HIT
U 1246, 0000,003C,0180,0A28,0200,1247  ;2997  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 1247, 0000,003C,0180,9000,0000,1248  ;2998  CACHE.INVALIDATE  ; MAKE SURE ITS NOT A HIT
U 1248, 0000,003C,0180,0803,0000,1249  ;2999  VA_VA+4
U 1249, 0000,003C,0180,9000,0000,108A  ;3000  CACHE.INVALIDATE  ; AND THE NEXT ONE TOO
U 108A, 0000,003D,0180,0800,0000,11DC  ;3001 =0 CALL,J/AUTOREFILLMISS  ; GO START THE AUTO REFILL
U 108B, 0000,003C,0180,C800,0000,124A  ;3002  D[LONG]_CACHE.P   ; EXECUTE THE CPU READ
U 124A, 1810,0038,01E0,0970,0000,124B  ;3003  IBC/STOP,Q_D,D_RC[0E]  ; SAVE RECEIVED DATA AND GET EXPECTED DATA
U 124B, 001D,2000,0180,0800,0010,124C  ;3004  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED DATA
U 124C, 0000,013C,0180,0800,0000,108C  ;3005  Z?    ; BRANCH IF OK
U 108C, 0001,203D,0180,09E8,0000,112E  ;3006 =0 ERROR2,RC[0D]_Q   ; CPU RECEIVED THE WRONG DATA
U 108D, 0000,003C,0180,0800,0000,108E  ;3007 END002: NOP
          ;3008
          ;3009
```

```
          ;3010 .PAGE "TEST 1EB AUTO REFILL AND ROM ENABLE 2 (READ.V.NEWPC)"
          ;3011 ;*************************************************************************
          ;3012 ;**
          ;3013 ;
          ;3014 ; AUTO REFILL AND ROM ENABLE 2 (READ.V.NEWPC)
          ;3015 ;
          ;3016 ; TEST DESCRIPTION
          ;3017 ;
          ;3018 ; THIS TEST CHECKS THAT A STALL OCCURS WHEN AUTO REFILL AND
          ;3019 ; ROM ENABLE 2 ARE BOTH ASSERTED. THIS IS DONE BY EXECUTING AN IB
          ;3020 ; AUTO REFILL HIT AT THE SAME TIME AS A READ.V.NEWPC AND CHECKING
          ;3021 ; THE SILO COUNTER FOR THE CORRECT NUMBER OF TICKS.
          ;3022 ;
          ;3023 ; LOGIC DESCRIPTION
          ;3024 ;
          ;3025 ;  DATA: EXPECTED SILO COUNTER
          ;3026 ;        RECEIVED SILO COUNTER
          ;3027 ;--
          ;3028 ;*************************************************************************
          ;3029 =0
U 108E, 0018,0039,0980,09F8,0000,1116  ;3030 T003: NEWTST[.2]
U 108F, 0000,003C,0180,0800,0000,1090  ;3031  NOP
U 1090, 0000,003D,0180,0800,0000,1036  ;3032 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 1091, 0000,003C,0180,0A28,0200,124D  ;3033  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 124D, 0818,0038,C180,0800,0000,124E  ;3034  D_K[.FFFF]   ; GENERATE DATA PATTERN TO WRITE
U 124E, 0802,403C,0180,0800,0000,124F  ;3035  D_D.SXT[WORD]   ; ...
U 124F, 0000,003C,0180,9800,0000,1250  ;3036  MCT/VALIDATE   ; WRITE THE DATA PATTERN
U 1250, 0000,003C,0180,0803,0000,1251  ;3037  VA_VA+4
U 1251, 0000,003C,0180,9800,0000,1092  ;3038  MCT/VALIDATE   ; ...
U 1092, 0000,003D,0180,0800,0000,112B  ;3039 =0 ERLOOP
U 1093, 0818,0038,5DF8,0800,0000,1252  ;3040  D_K[.7],Q_0   ; GENERATE EXPECTED SILO COUNTER
U 1252, 0000,003C,6580,0800,0084,7253  ;3041  SC_K[.10]
U 1253, 0D00,003C,0180,0800,0000,1254  ;3042  D_DAL.SC
U 1254, 0001,003C,0180,09F0,0000,1255  ;3043  RC[0E]_D   ; SAVE
U 1255, 0818,0038,7580,0800,0000,1256  ;3044  D_K[.20]   ; GENERATE PATTERN TO TURN SILO
U 1256, 0B00,003C,0180,0800,0000,1094  ;3045  D_D.SWAP   ; COUNTER ON
U 1094, 0000,003D,7180,3C00,0000,11D7  ;3046 =0 ID[COMP]_D,CALL,J/AUTOREFILLHIT ; GO EXECUTE THE AUTO REFILL
U 1095, 0000,003C,0180,6000,0000,1257  ;3047  MCT/READ.V.NEWPC  ; NOW THE ROM ENABLE 2
U 1257, 1001,003C,71F0,2E88,0000,1258  ;3048  Q_ID[COMP],IBC/STOP,R[1]_D ; READ THE SILO COUNTER
U 1258, 001C,0034,01C0,0A60,0000,1259  ;3049  Q_Q.AND.R[0C]   ; MASK
U 1259, 0810,0038,0180,0970,0000,125A  ;3050  D_RC[0E]   ; GET THE EXPECTED DATA
U 125A, 001D,2000,0180,0800,0010,125B  ;3051  ALU_Q-D,CLK.UBCC  ; CHECK
U 125B, 0000,013C,0180,0800,0000,1096  ;3052  Z?   ; BRANCH IF OK
U 1096, 0001,203D,0180,09E8,0000,112E  ;3053 =0 ERROR2,RC[0D]_Q   ; STALL DID NOT OCCUR
U 1097, 0000,003C,0180,0800,0000,1098  ;3054 END003: NOP
          ;3055
          ;3056
```

```
          ;3057 .PAGE "TEST 1EC AUTO REFILL HIT AND CPU WRITE"
          ;3058 ;*****************************************************************
          ;3059 ;++
          ;3060 ;
          ;3061 ; AUTO REFILL HIT AND CPU WRITE"
          ;3062 ;
          ;3063 ; TEST DESCRIPTION
          ;3064 ;
          ;3065 ; THIS TEST CHECKS THAT A STALL OCCURS IF A CPU WRITE IS EXECUTED
          ;3066 ; DURING AN AUTO REFILL CYCLE. THIS IS DONE BY USING THE SILO COUNTER
          ;3067 ; TO CHECK THE NUMBER OF CYCLES.
          ;3068 ;
          ;3069 ; LOGIC DESCRIPTION
          ;3070 ;
          ;3071 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
          ;3072 ;
          ;3073 ; ERROR DESCRIPTION
          ;3074 ;
          ;3075 ; DATA: EXPECTED SILO COUNTER
          ;3076 ;       RECEIVED SILO COUNTER
          ;3077 ;--
          ;3078 ;*****************************************************************
          ;3079 =0
U 1098, 0018.0039.0980.09F8.0000.1116  ;3080 T004: NEWTST[.2]
U 1099, 0000.003C.0180.0800.0000.109A  ;3081  NOP
U 109A, 0000.003D.0180.0800.0000.1036  ;3082 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 109B, 0000.003C.018D.0A28.0200.125C  ;3083  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 125C, 0818.0038.C180.0800.0000.125D  ;3084  D_K[.FFFF]   ; GENERATE DATA PATTERN TO WRITE
U 125D, 0802.403C.0180.0800.0000.125E  ;3085  D_D.SXT[WORD]   ; ...
U 125E, 0000.003C.0180.9800.0000.125F  ;3086  MCT/VALIDATE   ; WRITE THE DATA PATTERN
U 125F, 0000.003C.0180.0803.0000.1260  ;3087  VA_VA+4
U 1260, 0000.003C.0180.9800.0000.109C  ;3088  MCT/VALIDATE   ; ...
U 109C, 0000.003D.0180.0800.0000.112B  ;3089 =0 ERLOOP
U 109D, 0818.0038.5DF8.0800.0000.1261  ;3090  D_K[.7],Q_0   ; GENERATE EXPECTED SILO COUNTER
U 1261, 0000.003C.6580.0800.0084.7262  ;3091  SC_K[.10]
U 1262, 0D00.003C.0180.0800.0000.1263  ;3092  D_DAL.SC
U 1263, 0001.003C.0180.09F0.0000.1264  ;3093  RC[0E]_D   ; SAVE
U 1264, 0818.0038.7580.0800.0000.1265  ;3094  D_K[.20]   ; GENERATE PATTERN TO TURN SILO
U 1265, 0B00.003C.0180.0800.0000.109E  ;3095  D_D.SWAP   ; COUNTER ON
U 109E, 0000.003D.7180.3C00.0000.11D7  ;3096 =0 ID[COMP]_D,CALL,J/AUTOREFILLHIT ; GO EXECUTE THE AUTO REFILL
U 109F, 0000.003C.0180.A800.0000.1266  ;3097  CACHE.P_D[LONG]   ; NOW START THE CP WRITE
U 1266, 1001.003C.71F0.2E88.0000.1267  ;3098  Q_ID[COMP],IBC/STOP,R[1]_D ; READ THE SILO COUNTER
U 1267, 001C.0034.01C0.0A60.0000.1268  ;3099  Q_Q.AND.R[0C]   ; MASK
U 1268, 0810.0038.0180.0970.0000.1269  ;3100  D_RC[0E]   ; GET THE EXPECTED DATA
U 1269, 001D.2000.0180.0800.0010.126A  ;3101  ALU_Q-D,CLK.UBCC   ; CHECK
U 126A, 0000.013C.0180.0800.0000.10A0  ;3102  Z?   ; BRANCH IF OK
U 10A0, 0001.203D.0180.09E8.0000.112E  ;3103 =0 ERROR2,RC[0D]_Q   ; STALL DID NOT OCCUR
U 10A1, 0000.003C.0180.0800.0000.10A2  ;3104 END004: NOP
          ;3105
          ;3106
```

```
          ;3107 .PAGE "TEST 1ED AUTO REFILL AND CP VALIDATE"
          ;3108 ;*****************************************************************
          ;3109 ;++
          ;3110 ;
          ;3111 ; AUTO REFILL AND CP VALIDATE
          ;3112 ;
          ;3113 ; TEST DESCRIPTION
          ;3114 ;
          ;3115 ; THIS TEST CHECKS THAT A STALL OCCURS WHEN A VALIDATE FUNCTION IS
          ;3116 ; EXECUTED AT THE SAME TIME AN AUTO REFILL OCCURS.
          ;3117 ;
          ;3118 ; LOGIC DESCRIPTION
          ;3119 ;
          ;3120 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
          ;3121 ;
          ;3122 ; ERROR DESCRIPTION
          ;3123 ;
          ;3124 ; DATA: EXPECTED SILO COUNTER
          ;3125 ;       RECEIVED SILO COUNTER
          ;3126 ;--
          ;3127 ;*****************************************************************
          ;3128 =0
U 10A2, 0018.0039.0980.09F8.0000.1116  ;3129 T005: NEWTST[.2]
U 10A3, 0000.003C.0180.0800.0000.10A4  ;3130  NOP
U 10A4, 0000.003D.0180.0800.0000.1036  ;3131 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10A5, 0000.003C.0180.0A28.0200.126B  ;3132  ALU_R[5],VA_ALU   ; LOAD THE IB ADDRESS
U 126B, 0818.0038.C180.0800.0000.126C  ;3133  D_K[.FFFF]   ; GENERATE DATA PATTERN TO WRITE
U 126C, 0802.403C.0180.0800.0000.126D  ;3134  D_D.SXT[WORD]   ; ...
U 126D, 0000.003C.0180.9800.0000.126E  ;3135  MCT/VALIDATE   ; WRITE THE DATA PATTERN
U 126E, 0000.003C.0180.0B03.0000.126F  ;3136  VA_VA+4
U 126F, 0000.003C.0180.9800.0000.10A6  ;3137  MCT/VALIDATE   ; ...
U 10A6, 0000.003D.0180.0800.0000.112B  ;3138 =0 ERLOOP
U 10A7, 0818.0038.5DF8.0800.0000.1270  ;3139  D_K[.7],Q_0   ; GENERATE EXPECTED SILO COUNTER
U 1270, 0000.003C.6580.0800.0084.7271  ;3140  SC_K[.10]
U 1271, 0D00.003C.0180.0800.0000.1272  ;3141  D_DAL.SC
U 1272, 0001.003C.0180.09F0.0000.1273  ;3142  RC[0E]_D   ; SAVE
U 1273, 0818.0038.7580.0800.0000.1274  ;3143  D_K[.20]   ; GENERATE PATTERN TO TURN SILO
U 1274, 0B00.003C.0180.0800.0000.10A8  ;3144  D_D.SWAP   ; COUNTER ON
U 10A8, 0000.003D.7180.3C00.0000.11D7  ;3145 =0 ID[COMP]_D,CALL,J/AUTOREFILLHIT ; GO EXECUTE THE AUTO REFILL
U 10A9, 0000.003C.0180.9800.0000.1275  ;3146  MCT/VALIDATE   ; NOW EXECUTE THE VALIDATE CYCLE
U 1275, 1001.003C.71F0.2E88.0000.1276  ;3147  Q_ID[COMP],IBC/STOP,R[1]_D ; READ THE SILO COUNTER
U 1276, 001C.0034.01C0.0A60.0000.1277  ;3148  Q_Q.AND.R[0C]   ; MASK
U 1277, 0810.0038.0180.0970.0000.1278  ;3149  D_RC[0E]   ; GET THE EXPECTED DATA
U 1278, 001D.2000.0180.0800.0010.1279  ;3150  ALU_Q-D,CLK.UBCC   ; CHECK
U 1279, 0000.013C.0180.0800.0000.10AA  ;3151  Z?   ; BRANCH IF OK
U 10AA, 0001.203D.0180.09E8.0000.112E  ;3152 =0 ERROR2,RC[0D]_Q   ; STALL DID NOT OCCUR
U 10AB, 0000.003C.0180.0800.0000.10AC  ;3153 END005: NOP
          ;3154
          ;3155
```

```
                ;3156 .PAGE "TEST 1EE AUTO REFILL HIT WITH DISABLE SBI AND ISR"
                ;3157 ;*****************************************************************************
                ;3158 ;**
                ;3159 ;
                ;3160 ; AUTO REFILL HIT WITH DISABLE SBI AND ISR
                ;3161 ;
                ;3162 ; TEST DESCRIPTION
                ;3163 ;
                ;3164 ; THIS TEST CHECKS THAT A STALL OCCURS DURING AN AUTO REFILL HIT
                ;3165 ; WITH THE SBI DISABLED AND AN INTERRUPT SUMMERY READ IN THE SAME CYCLE
                ;3166 ; AS THE AUTO REFILL.
                ;3167 ;
                ;3168 ; LOGIC DESCRIPTION
                ;3169 ;
                ;3170 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
                ;3171 ;
                ;3172 ; ERROR DESCRIPTION
                ;3173 ;
                ;3174 ; DATA: EXPECTED SILO COUNTER
                ;3175 ;       RECEIVED SILO COUNTER
                ;3176 ;
                ;3177 ;--
                ;3178 ;*****************************************************************************
                ;3179 =0
U 10AC, 0018.0039.0980.09F8.0000.1116  ;3180 T006: NEWTST[.2]
U 10AD, 0000.003C.0180.0800.0000.10AE  ;3181  NOP
U 10AE, 0000.003D.0180.0800.0000.1036  ;3182 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10AF, 0818.0038.5DF8.0800.0000.127A  ;3183  D_K[.7],Q_0   ; GENERATE EXPECTED DATA
U 127A, 0000.003C.6580.0800.0084.727B  ;3184  SC_K[.10]
U 127B, 0D00.003C.0180.0800.0000.127C  ;3185  D_DAL.SC   ; ...
U 127C, 0001.003C.0180.09F0.0000.10B0  ;3186  RC[0E]_D   ; SAVE
U 10B0, 0000.003D.0180.0800.0000.112B  ;3187 =0 ERLOOP
U 10B1, 0F00.003C.0180.0A20.0200.127D  ;3188  ALU_R[4],VA_ALU,D_0  ; LOAD ADR OF IB
U 127D, 0000.003C.0180.9800.0000.127E  ;3189  MCT/VALIDATE   ; MAKE IT A HIT
U 127E, 0000.003C.0180.0803.0000.127F  ;3190  VA_VA+4
U 127F, 0000.003C.0180.9800.0000.1280  ;3191  MCT/VALIDATE   ; AND THE NEXT ADDRESS
U 1280, 0800.003C.0180.0A18.0000.1281  ;3192  D_R[3]    ; GET MAINTENANCE REG DATA
U 1281, 0000.003C.7580.3C00.0000.1282  ;3193  ID[MAINT]_D   ; DISABLE THE SBI
U 1282, 0818.0038.7580.0800.0000.1283  ;3194  D_K[.20]   ; SETUP TO START THE SILO COUNTER
U 1283, 0B00.003C.0180.0800.0000.10B2  ;3195  D_D.SWAP
U 10B2, 0000.003D.7180.3C00.0000.11D7  ;3196 =0 ID[COMP]_D,CALL,J/AUTOREFILLHIT ; START THE COUNTER AND AUTO REFILL
U 10B3, 0000.003C.0180.D800.0000.1284  ;3197  D_INT.SUM   ; EXECUTE THE INTERRUPT SUMMARY READ
U 1284, 1800.003C.71F0.2E30.0000.1285  ;3198  Q_ID[COMP],IBC/STOP,D_R[6] ; READ THE SILO COUNTER
U 1285, 0810.0038.7580.3D70.0000.1286  ;3199  ID[MAINT]_D,D_RC[0E]  ; GET THE EXPECTED DATA
U 1286, 001C.0034.01C0.0A60.0000.1287  ;3200  Q_Q.AND.R[0C]   ; MASK THE RECEIVED DATA
U 1287, 001D.2000.0180.0800.0010.1288  ;3201  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 1288, 0000.013C.0180.0800.0000.10B4  ;3202  Z?    ; BRANCH IF OK
U 10B4, 0001.203D.0180.09E8.0000.112E  ;3203 =0 ERROR2,RC[0D]_Q   ; FAILURE NOT DUE TO REFRESH
U 10B5, 0000.003C.0180.0800.0000.10B6  ;3204 END006: NOP
                ;3205
                ;3206
```

```
          ;3207 .PAGE "TEST 1EF AUTO REFILL HIT AND IB FLUSH"
          ;3208 ;*****************************************************************************
          ;3209 ;++
          ;3210 ;
          ;3211 ; AUTO REFILL HIT AND IB FLUSH
          ;3212 ;
          ;3213 ; TEST DESCRIPTION
          ;3214 ;
          ;3215 ; THIS TEST CHECKS THAT AN IB FLUSH DURING AN AUTO REFILL INHIBITS
          ;3216 ; THE SBI INTERFACE FROM GENERATEING THE STALL SIGNAL.
          ;3217 ; THIS IS CHECKED BY EXECUTING A CPU READ DURING THE FLUSH AND
          ;3218 ; MEASURING THE TIME IT TAKES TO COMPLETE.
          ;3219 ;
          ;3220 ; LOGIC DESCRIPTION
          ;3221 ;
          ;3222 ; THIS TEST CHECKS THE LOGIC ON THE TBM MODULE THAT CLEARS THE
          ;3223 ; "ENABLE IA" SIGNAL DURING AN IB FLUSH.
          ;3224 ;
          ;3225 ; ERROR DESCRIPTION
          ;3226 ;
          ;3227 ; DATA: EXPECTED SBI INTERFACE BUSY BIT
          ;3228 ;       RECEIVED SBI INTERFACE BUSY BIT
          ;3229 ;--
          ;3230 ;*****************************************************************************
          ;3231
          ;3232 =0
U 10B6, 0018,0039,0980,09F8,0000,1116  ;3233 T006A: NEWTST[.2]
U 10B7, 0000,003C,0180,0800,0000,10B8  ;3234  NOP
U 10B8, 0000,003D,0180,0800,0000,1036  ;3235 =0 CALL,J/MAINTREGSETUP
U 10B9, 0818,0038,D5F8,0800,0000,1289  ;3236  D_K[.6],Q_0   ; GENERATE THE EXPECTED SILO COUNTER
U 1289, 0000,003C,6580,0800,0084,728A  ;3237  SC_K[.10]
U 128A, 0D00,003C,0180,0800,0000,128B  ;3238  D_DAL.SC
U 128B, 0001,003C,0180,09F0,0000,128C  ;3239  RC[0E]_D
U 128C, 0F00,003C,0180,0A28,0200,128D  ;3240  VA_ALU,ALU_R[5],D_0   ; LOAD THE IB ADDRESS
U 128D, 0000,003C,0180,9800,0000,128E  ;3241  MCT/VALIDATE   ; MAKE IT A HIT
U 128E, 0000,003C,0180,0803,0000,128F  ;3242  VA_VA+4    ; AND THE NEXT ADDRESS
U 128F, 0000,003C,0180,9800,0000,1290  ;3243  MCT/VALIDATE
U 1290, 0000,003C,0180,0A20,0200,1291  ;3244  VA_ALU,ALU_R[4]   ; LOAD THE CPU ADDRESS
U 1291, 0000,003C,0180,9800,0000,10BA  ;3245  MCT/VALIDATE   ; MAKE IT A HIT
U 10BA, 0000,003D,0180,0800,0000,112B  ;3246 =0 ERLOOP
U 10BB, 0818,0038,7580,0800,0000,1292  ;3247  D_K[.20]   ; GENERATE PATTERN TO TURN THE SILO
U 1292, 0B00,003C,0180,0800,0000,1293  ;3248  D_D.SWAP   ; COUNTER ON
U 1293, 2000,003C,7180,3E28,4200,1294  ;3249  ID[COMP]_D,ALU_R[5],FLUSH.IB ; START THE COMPARATOR AND LOAD VA
U 1294, 0000,003C,0180,6000,0000,1295  ;3250  LOAD.IB    ; EXECUTE THE READ.V.NEWPC
U 1295, 3000,003C,0180,0800,0000,1296  ;3251  IBC/START   ; START THE IB
U 1296, 0000,003C,0180,F800,0000,1297  ;3252  MCT/ALLOW.IB.READ  ; ALLOW AN IB CYCLE
U 1297, 0000,003C,0180,0800,0000,1298  ;3253  NOP    ; IB OVER PAGE SHOULD OCCUR HERE
U 1298, 2000,003C,0180,F800,0000,1299  ;3254  MCT/ALLOW.IB.READ,IBC/FLUSH ; EXECUTE THE TEST FUNCTION
U 1299, 0000,003C,0180,C800,0000,129A  ;3255  D[LONG]_CACHE.P   ; EXECUTE A CPU READ TO CHECK THAT
U 129A, 1000,003C,71F0,2C00,0000,129B  ;3256  Q_ID[COMP],IBC/STOP  ; NO STALL OCCURS
U 129B, 001C,0034,01C0,0A60,0000,129C  ;3257  Q_Q.AND.R[0C]   ; MASK
U 129C, 0810,0038,0180,0970,0000,129D  ;3258  D_RC[0E]
U 129D, 001D,2000,0180,0800,0010,129E  ;3259  ALU_Q-D,CLK.UBCC  ; CHECK IF COUNT IS CORRECT
U 129E, 0000,013C,0180,0800,0000,10BC  ;3260  Z?
U 10BC, 0001,203D,0180,09E8,0000,112E  ;3261 =0 ERROR2,RC[0D]_Q   ; FLUSH DID NOT INHIBIT THE STALL
```

```
            ;3262 END006A:
U 10BD, 0000,003C,0180,0800,0000,10BE  ;3263  NOP
            ;3264
```

```
          ;3265 .PAGE "TEST 1F0 STALL<=TO TRP ENBLE*BUFFER FULL*DISABLE SBI*-AR OR SBI"
          ;3266 ;******************************************************************
          ;3267 ;++
          ;3268 ;
          ;3269 ; STALL<=TO TRP ENBLE*BUFFER FULL*DISABLE SBI*-AR OR SBI
          ;3270 ;
          ;3271 ; TEST DESCRIPTION
          ;3272 ;
          ;3273 ; THIS TEST CHECKS THAT STALL ASSERTS DURING AN INSTRUCTION BUFFER
          ;3274 ; READ MISS WITH DISABLE SBI CYCLE ON AND AN INTERRUPT SUMMARY READ.
          ;3275 ;
          ;3276 ; THIS IS DONE BY STARTING THE IB MISS, SETTING THE DISABLE SBI CYCLE
          ;3277 ; BIT, THEN EXECUTING AN INTERRUPT SUMMARY READ.
          ;3278 ;
          ;3279 ; LOGIC DESCRIPTION
          ;3280 ;
          ;3281 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE
          ;3282 ;
          ;3283 ; ERROR DESCRIPTION
          ;3284 ;
          ;3285 ; DATA: EXPECTED SILO COUNTER
          ;3286 ;       RECEIVED SILO COUNTER
          ;3287 ;--
          ;3288 ;******************************************************************
          ;3289 =0
U 10BE. 0018.0039.0980.09F8.0000.1116  ;3290 T007: NEWTST[.2]
U 10BF. 0000.003C.0180.0800.0000.1020  ;3291  NOP
U 1020. 0000.003D.0180.0800.0000.1036  ;3292 =00 CALL.J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 1021. 0818.0038.D980.0800.0000.1022  ;3293  D_K[.9]    ; GENERATE THE EXPECTED DATA
U 1022. 0000.003C.65F8.0800.0084.7023  ;3294  SC_K[.10],Q_0
U 1023. 0D00.003C.0180.0800.0000.129F  ;3295  D_DAL.SC
U 129F. 0001.003C.0180.09F0.0000.10C0  ;3296  RC[0E]_D    ; SAVE
U 10C0. 0000.003D.0180.0800.0000.112B  ;3297 =0 ERLOOP
U 10C1. 0000.003C.0180.0800.0000.10C2  ;3298  NOP
          ;3299 =0
U 10C2. 0000.003D.0180.0800.0000.1212  ;3300 RETRY2: CALL[WAITREFRESH]  ; WAIT FOR REFRESH TO OCCUR
U 10C3. 2F18.0038.1980.0800.4200.12A0  ;3301  ALU_0(K),FLUSH.IB,D_0  ; LOAD THE VA WITH THE IB ADDRESS
U 12A0. 0000.003C.0180.9000.0000.12A1  ;3302  CACHE.INVALIDATE  ; INVALIDATE IT
U 12A1. 0818.0038.7580.0800.0000.12A2  ;3303  D_K[.20]    ; SETUP TO START THE SILO COUNTER
U 12A2. 0B00.003C.0180.0800.0000.12A3  ;3304  D_D.SWAP    ; ...
U 12A3. 0000.003C.7180.3C00.0000.12A4  ;3305  ID[COMP]_D    ; START THE COUNTER
U 12A4. 3000.003C.0180.6000.0000.12A5  ;3306  LOAD.IB,IBC/START  ; START THE IB
U 12A5. 0800.003C.0180.FA18.0000.12A6  ;3307  MCT/ALLOW.IB.READ,D_R[3] ; GET MAINT REG DATA
U 12A6. 0000.003C.0180.F800.0000.12A7  ;3308  MCT/ALLOW.IB.READ  ; WAIT A CYCLE
U 12A7. 0000.003C.7580.3C00.0000.12A8  ;3309  ID[MAINT]_D    ; DISABLE THE SBI
U 12A8. 0000.003C.0180.D800.0000.12A9  ;3310  D_INT.SUM    ; START THE INTERRUPT SUMMARY READ
U 12A9. 1800.003C.71F0.2E30.0000.12AB  ;3311  Q_ID[COMP],IBC/STOP,D_R[6] ; READ THE SILO COUNTER
U 12AB. 0000.003C.7580.3C00.0000.12AC  ;3312  ID[MAINT]_D    ; LOAD ADR OF CONFIG REG B
U 12AC. 0000.003C.0180.C800.0000.12AD  ;3313  D[LONG]_CACHE.P    ; READ IT
U 12AD. 0001.003C.0180.0988.0000.12AE  ;3314  RC[1]_D    ; SAVE IT
U 12AE. 001C.0034.01C0.0A60.0000.12AF  ;3315  Q_Q.AND.R[0C]    ; MASK
U 12AF. 0810.0038.0180.0970.0000.12B0  ;3316  D_RC[0E]    ; GET THE EXPECTED DATA
U 12B0. 001D.2000.0180.0800.0010.12B1  ;3317  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 12B1. 0001.213C.0180.09E8.0000.10C4  ;3318  Z?,RC[0D]_Q    ; BRANCH IF OK
U 10C4. 0000.003D.0180.0800.0000.112E  ;3319 =0 ERROR2    ; FAILURE AND NO REFRESH
```

```
ZZ-ESKAR-V22  TEST 1F0 STALL<=TO TRP ENBLE*BUFFER FULL*DISABLE SBI*-AR OR SBI                    SEQ 1737
ESKAR51.MCR                    MICRO2  1P(00)    29-JUL-85  15:02:20                          Page   8

U 10C5, 0000,003C,0180,0800,0000,10C6  ;3320 END1: NOP
            ;3321
            ;3322
```

```
            ;3323 .PAGE "TEST 1F1 STALL DURING AN INTERRUPT SUMMARY READ"
            ;3324 ;*****************************************************************
            ;3325 ;**
            ;3326 ;
            ;3327 ; STALL DURING AN INTERRUPT SUMMARY READ
            ;3328 ;
            ;3329 ; TEST DESCRIPTION
            ;3330 ;
            ;3331 ; THIS TEST CHECKS THAT AN INTERRUPT SUMMARY READ STALLS FOR THE
            ;3332 ; CORRECT NUMBER OF CYCLES.
            ;3333 ;
            ;3334 ; LOGIC DESCRIPTION
            ;3335 ;
            ;3336 ; THIS TEST CHECKS PART OF THE STALL LOGIC ON THE SBL MODULE.
            ;3337 ;
            ;3338 ; ERROR DESCRIPTION
            ;3339 ;
            ;3340 ; DATA: EXPECTED SILO COUNTER
            ;3341 ;       RECEIVED SILO COUNTER
            ;3342 ;--
            ;3343 ;*****************************************************************
            ;3344 =0
U 10C6, 0018.0039.0980.09F8.0000.1116  ;3345 T008: NEWTST[.2]
U 10C7, 0000.003C.0180.0800.0000.10C8  ;3346  NOP
U 10C8, 0000.003D.0180.0800.0000.1036  ;3347 =0 CALL.J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10C9, 0818.0038.D5F8.0800.0000.12B2  ;3348  D_K[.6],Q_0   ; GENERATE THE EXPECTED DATA
U 12B2, 0000.003C.6580.0800.0084.72B3  ;3349  SC_K[.10]
U 12B3, 0D00.003C.0180.0800.0000.12B4  ;3350  D_DAL.SC
U 12B4, 0001.003C.0180.09F0.0000.10CA  ;3351  RC[0E]_D   ; SAVE
U 10CA, 0000.003D.0180.0800.0000.112B  ;3352 =0 ERLOOP
U 10CB, 0818.0038.7580.0800.0000.12B5  ;3353  D_K[.20]   ; GENERATE THE DATA TO START THE COUNTER
U 12B5, 0B00.003C.0180.0800.0000.12B6  ;3354  D_D.SWAP
U 12B6, 0000.003C.7180.3C00.0000.12B7  ;3355  ID[COMP]_D   ; START THE COUNTER
U 12B7, 0000.003C.0180.D800.0000.12B8  ;3356  D_INT.SUM   ; START THE INTRPT SUMM READ
U 12B8, 0000.003C.71F0.2C00.0000.12B9  ;3357  Q_ID[COMP]   ; READ THE COUNTER
U 12B9, 001C.0034.01C0.0A60.0000.12BA  ;3358  Q_Q.AND.R[0C]   ; MASK
U 12BA, 0810.0038.0180.0970.0000.12BB  ;3359  D_RC[0E]   ; GET THE EXPECTED DATA
U 12BB, 001D.2000.0180.0800.0010.12BC  ;3360  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 12BC, 0000.013C.0180.0800.0000.10CC  ;3361  Z?     ; BRANCH IF OK
U 10CC, 0001.203D.0180.09E8.0000.112E  ;3362 =0 ERROR2,RC[0D]_Q   ; FAILURE NOT DUE TO REFRESH
U 10CD, 0000.003C.0180.0800.0000.10CE  ;3363 END008: NOP
            ;3364
            ;3365
```

```
                ;3366 .PAGE "TEST 1F2 CPU WRITE MISS AND DATA MEMORY PARITY ERROR"
                ;3367 ;*****************************************************************
                ;3368 ;**
                ;3369 ;
                ;3370 ; CPU WRITE MISS AND DATA MEMORY PARITY ERROR
                ;3371 ;
                ;3372 ; TEST DESCRIPTION
                ;3373 ;
                ;3374 ; THIS TEST CHECKS THAT A CPU WRITE MISS WITH A DATA MEMORY PARITY
                ;3375 ; ERROR CAUSES BOTH GROUPS OF THE CACHE TO BE INVALIDATED.
                ;3376 ;
                ;3377 ; SUBTEST 1 - PARITY ERROR IN GROUP 0
                ;3378 ; SUBTEST 2 - PARITY ERROR IN GROUP 1
                ;3379 ;
                ;3380 ; LOGIC DESCRIPTION
                ;3381 ;
                ;3382 ; THIS TEST CHECKS THE LOGIC ON THE SBL MODULE THAT GENERATES A CACHE
                ;3383 ; INVALIDATE WHEN THE CPU DOES A WRITE WITH A CACHE PARITY ERROR.
                ;3384 ; IT ALSO CHECKS SOME LOGIC ON THE TBM MODULE THAT OR'S THE CACHE
                ;3385 ; WRITE PULSE SIGNALS.
                ;3386 ;
                ;3387 ; ERROR DESCRIPTION
                ;3388 ;
                ;3389 ; DATA: EXPECTED HIT/MISS BITS
                ;3390 ;       RECEIVED HIT/MISS BITS
                ;3391 ;--
                ;3392 ;*****************************************************************
                ;3393 =0
U 10CE, 0018,0039,0980,09F8,0000,1116  ;3394 T009: NEWTST[.2]
U 10CF, 0000,003C,0180,0800,0000,10D0  ;3395  NOP
U 10D0, 0000,003D,0180,0800,0000,1036  ;3396 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10D1, 0003,003C,0180,09F0,0000,10D2  ;3397  RC[0E]_0   ; SET THE EXPECTED DATA
U 10D2, 0000,003D,0180,0800,0000,1149  ;3398 =0 SUBTEST
U 10D3, 0000,003C,0180,0800,0000,10D4  ;3399  NOP
U 10D4, 0000,003D,0180,0800,0000,112B  ;3400 =0 ERLOOP
                ;3401
                ;3402 ;*
                ;3403 ; FIRST VALIDATE GROUP 1
                ;3404 ;-
                ;3405
U 10D5, 0018,0038,1980,0800,0200,12BD  ;3406  ALU_0(K),VA_ALU   ; LOAD THE ADDRESS
U 12BD, 0800,003C,0180,0A10,0000,12BE  ;3407  D_R[2]    ; GET MAINTENANCE REG DATA
U 12BE, 0F00,003C,7580,3C00,0000,12BF  ;3408  ID[MAINT]_D,D_0   ; LOAD THE MAINT REG
U 12BF, 0000,003C,0180,9800,0000,12C0  ;3409  MCT/VALIDATE   ; VALIDATE GROUP 1
                ;3410
                ;3411 ;*
                ;3412 ; NOW INVALIDATE GROUP 0 WITH A DATA PATTERN THAT WILL CAUSE A PARITY ERROR
                ;3413 ;-
                ;3414
U 12C0, 0800,003C,0180,0A30,0000,12C1  ;3415  D_R[6]    ; GET THE MAINT REG DATA
U 12C1, C000,003C,7580,3C00,0000,12C2  ;3416  ID[MAINT]_D   ; LOAD THE MAINT REG
U 12C2, 0818,0038,0580,0800,0000,12C3  ;3417  D_K[.1]    ; GET DATA TO WRITE
U 12C3, 0000,003C,0180,9000,0000,12C4  ;3418  CACHE.INVALIDATE  ; INVALIDATE GROUP 0
                ;3419
                ;3420 ;*
```

```
            ;3421 ; NOW SETUP THE MAINT REG TO FORCE THE PARITY ERROR
            ;3422 ;-
            ;3423
U 12C4, 0800,003C,0180,0A50,0000,12C5  ;3424  D_R[0A]    ; GET THE MAINT REG DATA
U 12C5, 0F00,003C,7580,3C00,0000,12C6  ;3425  ID[MAINT]_D,D_0   ; LOAD THE FORCE PARITY ERROR BITS
U 12C6, 0000,003C,0180,0800,0000,12C7  ;3426  NOP
U 12C7, 0000,003C,0180,A800,0000,12C8  ;3427  CACHE.P_D[LONG]   ; EXECUTE THE CPU WRITE
U 12C8, 0800,003C,0180,0A30,0000,12C9  ;3428  D_R[6]    ; CLEAR PARITY INVERT BITS
U 12C9, 0000,003C,7580,3C00,0000,12CA  ;3429  ID[MAINT]_D   ; ...
            ;3430
            ;3431 ;*
            ;3432 ; NOW CHECK THAT GROUP 1 WAS WRITTEN INVALID
            ;3433 ;-
            ;3434
U 12CA, 0800,003C,0180,0A10,0000,12CB  ;3435  D_R[2]    ; GET MAINT REG DATA TO FORCE G1
U 12CB, 0000,003C,7580,3C00,0000,12CC  ;3436  ID[MAINT]_D
U 12CC, 0000,003C,0180,0800,0000,12CD  ;3437  NOP
U 12CD, 0000,003C,0180,C800,0000,12CE  ;3438  D[LONG]_CACHE.P   ; READ FROM GROUP 1
U 12CE, 0000,003C,75F0,2C00,0000,12CF  ;3439  Q_ID[MAINT]   ; GET THE HIT/MISS BITS
U 12CF, 001C,0034,01C0,0A38,0010,12D0  ;3440  Q_Q.AND.R[7],CLK.UBCC   ; MASK
U 12D0, 0000,013C,0180,0800,0000,10D6  ;3441  Z?    ; BRANCH IF OK
U 10D6, 0001,203D,0180,09E8,0000,112E  ;3442 =0 ERROR2,RC[0D]_Q   ; GROUP 1 WAS NOT INVALIDATED.
            ;3443
            ;3444
            ;3445 ;////////////////////////////////////////////////////////////////////////
            ;3446 ;*
            ;3447 ; NOW CHECK A CPU WRITE MISS AND DATA MEMORY PARITY ERROR IN GROUP 1
            ;3448 ;-
            ;3449
U 10D7, 0000,003C,0180,0800,0000,10D8  ;3450  NOP
U 10D8, 0000,003D,0180,0800,0000,1149  ;3451 =0 SUBTEST
U 10D9, 0000,003C,0180,0800,0000,10DA  ;3452  NOP
U 10DA, 0000,003D,0180,0800,0000,112B  ;3453 =0 ERLOOP
            ;3454
            ;3455 ;*
            ;3456 ; FIRST VALIDATE GROUP 0
            ;3457 ;-
            ;3458
U 10DB, 0800,003C,0180,0A30,0000,12D1  ;3459  D_R[6]    ; GET MAINT REG DATA
U 12D1, 0F00,003C,7580,3C00,0000,12D2  ;3460  ID[MAINT]_D,D_0
U 12D2, 0000,003C,0180,9800,0000,12D3  ;3461  MCT/VALIDATE   ; VALIDATE GROUP 0
            ;3462
            ;3463 ;*
            ;3464 ; NOW INVALIDATE GROUP 1
            ;3465 ;-
            ;3466
U 12D3, 0800,003C,0180,0A10,0000,12D4  ;3467  D_R[2]    ; GET MAINT REG DATA
U 12D4, 0000,003C,7580,3C00,0000,12D5  ;3468  ID[MAINT]_D
U 12D5, 0818,0038,0580,0800,0000,12D6  ;3469  D_K[.1]
U 12D6, 0000,003C,0180,9000,0000,12D7  ;3470  CACHE.INVALIDATE
            ;3471
            ;3472 ;*
            ;3473 ; NOW SETUP TO FORCE THE PARITY ERROR
            ;3474 ;-
            ;3475
```

```
U 12D7, 0818.0038.7D80.0800.0000.12D8  ;3476  D_K[.18]    ; GENERATE THE REVERSE PARITY BITS
U 12D8, 0819.0000.09F8.0800.0000.12D9  ;3477  D_D-K[.2],Q_0    ; ...
U 12D9, 0000.003C.6580.0800.0084.72DA  ;3478  SC_K[.10]
U 12DA, 0D00.003C.0180.0800.0000.12DB  ;3479  D_DAL.SC    ; D=160000
U 12DB, 081C.2030.0180.0A10.0000.12DC  ;3480  D_D.OR.R[2]    ; OR WITH FORCE REPLACE BITS
U 12DC, 0000.003C.7580.3C00.0000.12DD  ;3481  ID[MAINT]_D
U 12DD, 0000.003C.0180.0800.0000.12DE  ;3482  NOP
          ;3483
          ;3484 ;*
          ;3485 ; NOW EXECUTE THE WRITE
          ;3486 ;-
          ;3487
U 12DE, 0000.003C.0180.A800.0000.12DF  ;3488  CACHE.P_D[LONG]
U 12DF, 0800.003C.0180.0A10.0000.12E0  ;3489  D_R[2]    ; CLEAR THE PARITY INVERT BITS
U 12E0, 0000.003C.7580.3C00.0000.12E1  ;3490  ID[MAINT]_D    ; ...
          ;3491
          ;3492 ;*
          ;3493 ; NOW CHECK THAT GROUP 0 WAS INVALIDATED
          ;3494 ;-
          ;3495
U 12E1, 0800.003C.0180.0A30.0000.12E2  ;3496  D_R[6]
U 12E2, 0000.003C.7580.3C00.0000.12E3  ;3497  ID[MAINT]_D    ; LOAD FORCE MISS BIT
U 12E3, 0000.003C.0180.0800.0000.12E4  ;3498  NOP
U 12E4, 0000.003C.0180.C800.0000.12E5  ;3499  D[LONG]_CACHE.P    ; READ FROM GROUP 0
U 12E5, 0000.003C.75F0.2C00.0000.12E6  ;3500  Q_ID[MAINT]    ; READ THE HIT/MISS BITS
U 12E6, 001C.0034.01C0.0A38.0010.12E7  ;3501  Q_Q.AND.R[7],CLK.UBCC    ; MASK AND CHECK
U 12E7, 0000.013C.0180.0800.0000.10DC  ;3502  Z?    ; BRANCH IF OK
U 10DC, 0001.203D.0180.09E8.0000.112E  ;3503 =0 ERROR2,RC[0D]_Q    ; GROUP 0 WAS NOT INVALIDATED
U 10DD, 0000.003C.0180.0800.0000.10DE  ;3504 END009: NOP
          ;3505
```

```
            ;3506 .PAGE "TEST 1F3 CPU WRITE HIT AND CACHE DATA PARITY ERROR"
            ;3507 ;*****************************************************************
            ;3508 ;+*
            ;3509 ;
            ;3510 ; CPU WRITE HIT AND CACHE DATA PARITY ERROR
            ;3511 ;
            ;3512 ; TEST DESCRIPTION
            ;3513 ;
            ;3514 ; THIS TEST CHECKS THAT AN INVALIDATE CYCLE IS NOT EXECUTED IF
            ;3515 ; A CPU WRITE AND DATA PARITY ERROR ALSO IS A HIT.
            ;3516 ;
            ;3517 ; LOGIC DESCRIPTION
            ;3518 ;
            ;3519 ; THIS TEST CHECKS THE CACHE WRITE PULSE MULTIPLEXOR ON THE TBM
            ;3520 ; MODULE AND THE INVALIDATE CIRCUIT ON THE SBL MODULE.
            ;3521 ;
            ;3522 ; ERROR DESCRIPTION
            ;3523 ;
            ;3524 ; DATA: EXPECTED HIT/MISS BITS
            ;3525 ;       RECEIVED HIT/MISS BITS
            ;3526 ;--
            ;3527 ;*****************************************************************
            ;3528 =0
U 10DE, 0018,0039,0980,09F8,0000,1116  ;3529 T00A: NEWTST[.2]
U 10DF, 0000,003C,0180,0800,0000,10E0  ;3530  NOP
U 10E0, 0000,003D,0180,0800,0000,1036  ;3531 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10E1, 0818,0038,0980,0800,0000,12E8  ;3532  D_K[.2]    ; GENERATE EXPECTED DATA
U 12E8, 0000,003C,01F8,0800,0084,72E9  ;3533  SC_K[.8],Q_0   ; ...
U 12E9, 0D00,003C,0180,0800,0000,12EA  ;3534  D_DAL.SC    ; D=200
U 12EA, 0001,003C,0180,09F0,0000,10E2  ;3535  RC[0E]_D
U 10E2, 0000,003D,0180,0800,0000,1149  ;3536 =0 SUBTEST
U 10E3, 0000,003C,0180,0800,0000,10E4  ;3537  NOP
U 10E4, 0000,003D,0180,0800,0000,112B  ;3538 =0 ERLOOP
            ;3539
            ;3540 ;+
            ;3541 ; MAKE GROUP 0 A HIT WITH A PATTERN THAT WILL CAUSE A PARITY ERROR
            ;3542 ;-
            ;3543
U 10E5, 0018,0038,1980,0800,0200,12EB  ;3544  ALU_0(K),VA_ALU   ; LOAD THE ADDRESS
U 12EB, 0818,0038,0580,0800,0000,12EC  ;3545  D_K[.1]
U 12EC, 0000,003C,0180,9800,0000,12ED  ;3546  MCT/VALIDATE   ; WRITE TO GROUP 0
            ;3547
            ;3548 ;+
            ;3549 ; SETUP TO DO THE WRITE WITH A PARITY ERROR
            ;3550 ;-
            ;3551
U 12ED, 0800,003C,0180,0A50,0000,12EE  ;3552  D_R[0A]
U 12EE, 0000,003C,7580,3C00,0000,12EF  ;3553  ID[MAINT]_D   ; LOAD PARITY INVERT BITS
U 12EF, 0818,0038,0580,0800,0000,12F0  ;3554  D_K[.1]
U 12F0, 0000,003C,0180,A800,0000,12F1  ;3555  CACHE.P_D[LONG]   ; EXECUTE THE WRITE
U 12F1, 0800,003C,0180,0A30,0000,12F2  ;3556  D_R[6]
U 12F2, 0000,003C,7580,3C00,0000,12F3  ;3557  ID[MAINT]_D   ; CLEAR PARITY INVERT BITS
            ;3558
            ;3559 ;+
            ;3560 ; CHECK THAT GROUP 0 IS STILL A HIT
```

```
            ;3561 ;-
            ;3562
U 12F3, 0000.003C.0180.0800.0000.12F4  ;3563  NOP
U 12F4, 0000.003C.0180.C800.0000.12F5  ;3564  D[LONG]_CACHE.P    ; EXECUTE A READ
U 12F5, 0000.003C.75F0.2C00.0000.12F6  ;3565  Q_ID[MAINT]    ; READ THE HIT/MISS BITS
U 12F6, 0810.0038.0180.0970.0000.12F7  ;3566  D_RC[0E]    ; GET THE EXPECTED DATA
U 12F7, 001C.0034.01C0.0A38.0000.12F8  ;3567  Q_Q.AND.R[7]    ; MASK Q
U 12F8, 001D.2000.0180.0800.0010.12F9  ;3568  ALU_Q-D,CLK.UBCC   ; CHECK
U 12F9, 0000.013C.0180.0800.0000.10E6  ;3569  Z?     ; BRANCH IF OK
U 10E6, 0001.203D.0180.09E8.0000.112E  ;3570 =0 ERROR2,RC[0D]_Q    ; INVALIDATE CYCLE MUST HAVE OCCURRED
            ;3571
            ;3572
            ;3573 ;////////////////////////////////////////////////////////////////////////////
            ;3574 ;*
            ;3575 ; NOW CHECK GROUP 1
            ;3576 ;-
            ;3577
U 10E7, 0000.003C.0180.0800.0000.10E8  ;3578  NOP
U 10E8, 0000.003D.0180.0800.0000.1149  ;3579 =0 SUBTEST
U 10E9, 0818.0038.1180.0800.0000.12FA  ;3580  D_K[.4]     ; GENERATE EXPECTED DATA
U 12FA, 0000.003C.01F8.0800.0084.72FB  ;3581  SC_K[.8],Q_0    ; SETUP THE SHIFT COUNT
U 12FB, 0D00.003C.0180.0800.0000.12FC  ;3582  D_DAL.SC    ; ...
U 12FC, 0001.003C.0180.09F0.0000.10EA  ;3583  RC[0E]_D    ; SAVE EXPECTED DATA
U 10EA, 0000.003D.0180.0800.0000.112B  ;3584 =0 ERLOOP
U 10EB, 0800.003C.0180.0A10.0000.12FD  ;3585  D_R[2]     ; SETUP TO VALIDATE GROUP 1
U 12FD, 0000.003C.7580.3C00.0000.12FE  ;3586  ID[MAINT]_D
U 12FE, 0818.0038.0580.0800.0000.12FF  ;3587  D_K[.1]
U 12FF, 0000.003C.0180.9800.0000.1300  ;3588  MCT/VALIDATE    ; VALIDATE GROUP 1
            ;3589
            ;3590 ;*
            ;3591 ; SETUP TO FORCE A PARITY ERROR IN GROUP 1
            ;3592 ;-
            ;3593
U 1300, 0818.0038.7DF8.0800.0000.1301  ;3594  D_K[.18],Q_0
U 1301, 0819.0000.0980.0800.0000.1302  ;3595  D_D-K[.2]
U 1302, 0000.003C.6580.0800.0084.7303  ;3596  SC_K[.10]
U 1303, 0D00.003C.0180.0800.0000.1304  ;3597  D_DAL.SC    ; D=160000
U 1304, 081C.2030.0180.0A10.0000.1305  ;3598  D_D.OR.R[2]    ; INSERT FORCE MISS BIT
U 1305, 0000.003C.7580.3C00.0000.1306  ;3599  ID[MAINT]_D    ;
            ;3600
            ;3601 ;*
            ;3602 ; NOW EXECUTE THE WRITE
            ;3603 ;-
            ;3604
U 1306, 0818.0038.0580.0800.0000.1307  ;3605  D_K[.1]
U 1307, 0000.003C.0180.A800.0000.1308  ;3606  CACHE.P_D[LONG]    ; EXECUTE THE WRITE
U 1308, 0800.003C.0180.0A10.0000.1309  ;3607  D_R[2]     ; CLEAR THE INVERT PARITY BITS
U 1309, 0000.003C.7580.3C00.0000.130A  ;3608  ID[MAINT]_D    ; ...
            ;3609
            ;3610 ;*
            ;3611 ; NOW CHECK THAT GROUP 1 IS STILL VALID
            ;3612 ;-
            ;3613
U 130A, 0000.003C.0180.0800.0000.130B  ;3614  NOP
U 130B, 0000.003C.0180.C800.0000.130C  ;3615  D[LONG]_CACHE.P    ; READ FROM GROUP 1
```

```
U 130C, 0000,003C,75F0,2C00,0000,130D  ;3616  Q_ID[MAINT]   ; READ THE HIT MISS BITS
U 130D, 0810,0038,0180,0970,0000,130E  ;3617  D_RC[0E]
U 130E, 001C,0034,01C0,0A38,0000,130F  ;3618  Q_Q.AND.R[7]   ; MASK
U 130F, 001D,2000,0180,0800,0010,1310  ;3619  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED HIT/MISS BITS
U 1310, 0000,013C,0180,0800,0000,10EC  ;3620  Z?    ; BRANCH IF OK
U 10EC, 0001,203D,0180,09E8,0000,112E  ;3621 =0 ERROR2,RC[0D]_Q   ; GROUP 1 MUST HAVE BEEN INVALIDATED
U 10ED, 0000,003C,0180,0800,0000,10EE  ;3622 END00A: NOP
            ;3623
            ;3624
```

```
            ;3625 .PAGE "TEST 1F4 MASK ON INSTRUCTION BUFFER READS"
            ;3626 ;*****************************************************************
            ;3627 ;++
            ;3628 ;
            ;3629 ; MASK ON INSTRUCTION BUFFER READS
            ;3630 ;
            ;3631 ; TEST DESCRIPTION
            ;3632 ;
            ;3633 ; THIS TEST EXECUTES AN INSTRUCTION BUFFER READ TO EITHER A UBA OR
            ;3634 ; MBA CONTROL REGISTER. IF THE BYTE MASK IS NOT SENT OUT AS ALL ONE'S
            ;3635 ; THE REFERANCE WILL TIMEOUT. THIS IS CHECKED BY EXAMINING THE TIMEOUT
            ;3636 ; BIT IN THE SBI ERROR REGISTER.
            ;3637 ;
            ;3638 ; LOGIC DESCRIPTION
            ;3639 ;
            ;3640 ; THIS TEST CHECKS THE SIGNAL THAT CONTROLS THE BYTE MASK MUX ON
            ;3641 ; THE SBH MODULE.
            ;3642 ;
            ;3643 ; ERROR DESCRIPTION
            ;3644 ;
            ;3645 ; DATA: EXPECTED SBI ERROR REGISTER
            ;3646 ;       RECEIVED SBI ERROR REGISTER
            ;3647 ;--
            ;3648 ;*****************************************************************
            ;3649 =0
U 10EE, 0018.0039.0980.09F8.0000.1116  ;3650 T00B: NEWTST[.2]
U 10EF, 0000.003C.0180.0B00.0000.10F0  ;3651  NOP
U 10F0, 0000.003D.0180.0B00.0000.1036  ;3652 =0 CALL,J/MAINTREGSETUP  ; SETUP THE REGISTERS
U 10F1, 0018.0038.0980.09F0.0000.10F2  ;3653  RC[0E]_K[.2]   ; SET THE EXPECTED DATA
U 10F2, 0000.003D.0180.0B00.0000.11E9  ;3654 =0 CALL,J/LOOKFORDEVICE  ; SEE IF THERE IS A DEVICE ON THE SBI
U 10F3, 0000.003C.0180.0A00.0010.1311  ;3655  ALU_R[0],CLK.UBCC  ; WAS THERE A DEVICE?
U 1311, 0000.013C.0180.0B00.0000.10F4  ;3656  Z?    ; BRANCH IF NO
U 10F4, 0000.003C.0180.0B00.0000.10F6  ;3657 =0 J/T00BS1
U 10F5, 0000.003C.0180.0800.0000.10FB  ;3658  J/T00BEND
            ;3659 =0
            ;3660 T00BS1:
U 10F6, 0000.003D.0180.0800.0000.112B  ;3661  ERLOOP
U 10F7, 0818.0038.3180.0800.0000.1312  ;3662  D_K[.40]   ; SETUP TO INIT THE ERROR REGISTER
U 1312, 0000.003C.6580.3C00.0000.1313  ;3663  ID[SBI.ERR]_D   ; CLEAR THE ERROR REGISTER
U 1313, 2000.003C.0180.0A00.4200.10F8  ;3664  ALU_R[0],FLUSH.IB  ; LOAD THE VIBA
U 10F8, 3000.003D.0180.6000.0000.11D3  ;3665 =0 LOAD.IB,IBC/START,CALL,J/WAITFORTIMEOUT ; START THE IB REFERANCE
U 10F9, 1818.0038.0980.0800.0000.1314  ;3666  IBC/STOP,D_K[.2]  ; GET EXPECTED SBI ERROR REG DATA
U 1314, 0000.003C.65F0.2C00.0000.1315  ;3667  Q_ID[SBI.ERR]   ; READ THE ERROR REGISTER
U 1315, 001D.2000.0180.0800.0010.1316  ;3668  ALU_Q-D,CLK.UBCC  ; DID A TIMEOUT OCCUR?
U 1316, 0000.013C.0180.0800.0000.10FA  ;3669  Z?    ; BRANCH IF NO
U 10FA, 0001.203D.0180.09E8.0000.112E  ;3670 =0 ERROR2,RC[0D]_Q   ; MASK ON IB READ INCORRECT
U 10FB, 0000.003C.0180.0800.0000.10FC  ;3671 T00BEND:NOP
            ;3672
            ;3673
```

```
            ;3674 .PAGE "TEST 1F5 AUTO REFILL CACNELED BY INSTRUCTION BUFFER FLUSH"
            ;3675 ;*****************************************************************
            ;3676 ;**
            ;3677 ;
            ;3678 ; AUTO REFILL CANCELED BY INSTRUCTION BUFFER FLUSH
            ;3679 ;
            ;3680 ; TEST DESCRIPTION
            ;3681 ;
            ;3682 ; THIS TEST CHECKS THAT AN AUTO REFILL CYCLE IS CANCELED IF AN
            ;3683 ; INSTRUCTION BUFFER FLUSH OCCURS IN THE SAME CYCLE. THIS IS DONE
            ;3684 ; BY CHECKING THAT THE AUTO REFILL STALL DOES NOT OCCUR.
            ;3685 ;
            ;3686 ; LOGIC DESCRIPTION
            ;3687 ;
            ;3688 ; THIS TEST CHECKS THE LOGIC ON THE TBM MODULE THAT INHIBITS THE
            ;3689 ; SIGNAL "ENABLE IA".
            ;3690 ;
            ;3691 ; ERROR DESCRIPTION
            ;3692 ;
            ;3693 ; DATA: EXPECTED SILO COUNT
            ;3694 ;       RECEIVED SILO COUNT
            ;3695 ;--
            ;3696 ;*****************************************************************
            ;3697
            ;3698 =0
U 10FC. 0018.0039.0980.09F8.0000.1116  ;3699 T00C: NEWTST[.2]
U 10FD. 0000.003C.0180.0800.0000.1024  ;3700  NOP
U 1024. 0000.003D.0180.0800.0000.1036  ;3701 =00 CALL,J/MAINTREGSETUP  ; SETUP SOME REGISTERS
U 1025. 0000.003C.0180.0800.0000.1026  ;3702  NOP
U 1026. 0000.003C.0180.0A20.0200.1317  ;3703  ALU_R[4],VA_ALU   ; LOAD TEH CPU ADDRESS
U 1317. 0000.003C.0180.9800.0000.1318  ;3704 = MCT/VALIDATE    ; MAKE IT A HIT
U 1318. 0000.003C.0180.0803.0000.1319  ;3705  VA_VA+4     ; AND THE NEXT LONG WORD
U 1319. 0000.003C.0180.9800.0000.131A  ;3706  MCT/VALIDATE
U 131A. 0000.003C.0180.0A28.0200.131B  ;3707  VA_ALU,ALU_R[5]    ; LOAD THE IB ADDRESS
U 131B. 0000.003C.0180.9800.0000.131C  ;3708  MCT/VALIDATE    ; MAKE IT A HIT
U 131C. 0000.003C.0180.0803.0000.131D  ;3709  VA_VA+4     ; AND THE NEXT LONG WORD
U 131D. 0000.003C.0180.9800.0000.131E  ;3710  MCT/VALIDATE
U 131E. 0818.0038.5DF8.0800.0000.131F  ;3711  D_K[.7],Q_0    ; GENERATE THE EXPECTED SILO COUNTER
U 131F. 0000.003C.6580.0800.0084.7320  ;3712  SC_K[.10]
U 1320. 0D00.003C.0180.0800.0000.1321  ;3713  D_DAL.SC
U 1321. 0001.003C.0180.09F0.0000.10FE  ;3714  RC[0E]_D    ; SAVE
U 10FE. 0000.003D.0180.0800.0000.112B  ;3715 =0 ERLOOP
U 10FF. 0818.0038.7580.0800.0000.1322  ;3716  D_K[.20]    ; GENERATE DATA TO LOAD SILO REGISTER
U 1322. 0B00.003C.0180.0800.0000.110A  ;3717  D_D.SWAP
U 110A. 0000.003D.7180.3C00.0000.11D7  ;3718 =0 ID[COMP]_D,CALL[AUTOREFILLHIT] ; START THE AUTO REFILL
U 110B. 2000.003C.0180.C800.0000.1323  ;3719  IBC/FLUSH,D[LONG]_CACHE.P ; START CPU READ AND FLUSH
U 1323. 1000.003C.71F0.2C00.0000.1324  ;3720  IBC/STOP,Q_ID[COMP]  ; STOP IB AND GET THE COUNT
U 1324. 001C.0034.01C0.0A60.0000.1325  ;3721  Q_Q.AND.R[0C]    ; MASK SILO REGISTER DATA
U 1325. 0810.0038.0180.0970.0000.1326  ;3722  D_RC[0E]    ; GET EXPECTED DATA
U 1326. 001D.2000.0180.0800.0010.1327  ;3723  ALU_Q-D,CLK.UBCC   ; NO STALL?
U 1327. 0001.213C.0180.09E8.0000.1110  ;3724  Z?,RC[0D]_Q    ; BRANCH IF YES
U 1110. 0000.003D.0180.0800.0000.112E  ;3725 =0 ERROR2     ; CPU STALLED
U 1111. 0000.003C.0180.0800.0000.1112  ;3726  NOP
            ;3727
```

```
            ;3728  .PAGE "TEST 1F6 RESERVED"
            ;3729 =0
U 1112. 0000,003D,0180,0800,0000,1116  ;3730 T1F6: NEWTST
U 1113. 0000,003C,0180,0800,0000,1328  ;3731  NOP
            ;3732
```

```
            ;3733  .PAGE "DUMMY NEWTST"
U 1328. 0000,003D,0180,0800,0000,1116  ;3734 DUM: NEWTST
            ;3735
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1918: 1887 1889= 1908= 2515= 2516= 2517= 1888
U 1008  2326= 2327= 2328= 2329= 2330= 2331= 2332= 2333=
U 1010  2523= 2524= 2526= 1911 2530= 2531= 2532= 1912
U 1018  2536= 2539= 2541= 2542= 2748= 2749= 2750= 2751=
U 1020  3292= 3293= 3294= 3295= 3701= 3702= 3703= 1913
U 1028  1909= 1910= 1926= 1927= 1981= 1983= 1994= 1995=
U 1030  2125= 2127= 2130= 2132= 2135= 2137= 2162= 2163=
U 1038  2164= 2165= 2174= 2175= 2176= 2177= 2178= 2179=
U 1040  2259= 2260= 2302= 2303= 2304= 2306= 2314= 2315=
U 1048  2316= 2318= 2321= 2322= 2363= 2364= 2000: 1914
U 1050  2365= 2366= 2373= 2374= 2380= 2382= 2385= 2386=
U 1058  2405= 2406= 2004: 1915 2408= 2409= 2008: 1916
U 1060  2582= 2583= 2615= 2616= 2619= 2620= 2631= 2632=
U 1068  2662= 2663= 2725= 2727= 2736= 2737= 2815= 2816=
U 1070  2818= 2819= 2850= 2851= 2889= 2890= 2891= 2892=
U 1078  2902= 2903= 2909= 2910= 2917= 2923= 2928= 2930=
U 1080  2942= 2948= 2955= 2956= 2982= 2983= 2984= 2985=
U 1088  2994= 2995= 3001= 3002= 3006= 3007= 3030= 3031=
U 1090  3032= 3033= 3039= 3040= 3046= 3047= 3053= 3054=
U 1098  3080= 3081= 3082= 3083= 3089= 3090= 3096= 3097=
U 10A0  3103= 3104= 3129= 3130= 3131= 3132= 3138= 3139=
U 10A8  3145= 3146= 3152= 3153= 3180= 3181= 3182= 3183=
U 10B0  3187= 3188= 3196= 3197= 3203= 3204= 3233= 3234=
U 10B8  3235= 3236= 3246= 3247= 3261= 3263= 3290= 3291=
U 10C0  3297= 3298= 3300= 3301= 3319= 3320= 3345= 3346=
U 10C8  3347= 3348= 3352= 3353= 3362= 3363= 3394= 3395=
U 10D0  3396= 3397= 3398= 3399= 3400= 3406= 3442= 3450=
U 10D8  3451= 3452= 3453= 3459= 3503= 3504= 3529= 3530=
U 10E0  3531= 3532= 3536= 3537= 3538= 3544= 3570= 3578=
U 10E8  3579= 3580= 3584= 3585= 3621= 3622= 3650= 3651=
U 10F0  3652= 3653= 3654= 3655= 3657= 3658= 3661= 3662=
U 10F8  3665= 3666= 3670= 3671= 3699= 3700= 3715= 3716=
U 1100  1873: 1874: 1875: 1876: 1877: 1878: 1879: 1880:
U 1108  1881: 1917 3718= 3719= 1882: 1883: 1884: 1885:
U 1110  3725= 3726= 3730= 3731= 1924 1925 1970 1971
U 1118  1972 1973 1974 1975 1976 1977 1978 1979
U 1120  1980 1984 1985 1986 1987 1988 1989 1990
U 1128  1991 1992 1993 1997 1998 1999 2002 2003
U 1130  2027 2028 2029 2030 2068 2069 2070 2071
U 1138  2072 2073 2074 2075 2076 2077 2078 2079
U 1140  2080 2081 2082 2083 2085 2086 2087 2088
U 1148  2090 2100 2101 2109 2110 2111 2120 2121
U 1150  2122 2123 2124 2128 2129 2112: 2133 2134
U 1158  2138 2139 2166 2167 2171 2173 2180 2181
U 1160  2182 2183 2184 2185 2186 2187 2188 2189
U 1168  2190 2191 2192 2193 2194 2195 2196 2197
U 1170  2198 2199 2200 2201 2202 2203 2204 2205
U 1178  2206 2234 2235 2236 2261 2262 2263 2264
U 1180  2307 2308 2309 2310 2319 2320 2324 2325
U 1188  2338 2339 2341 2342 2343 2345 2350 2351
U 1190  2352 2354 2359 2361 2362 2370 2371 2375
U 1198  2376 2377 2378 2379 2387 2388 2389 2391
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2392 2393 2394 2395 2396 2401 2403 2404
U 11A8  2407 2432 2433 2434 2435 2464 2465 2467
U 11B0  2468 2469 2513 2514 2543 2571 2572 2573
U 11B8  2574 2575 2576 2577 2578 2580 2581 2611
U 11C0  2613 2614 2618 2622 2623 2624 2625 2627
U 11C8  2628 2629 2630 2633 2634 2635 2646 2647
U 11D0  2648 2649 2650 2658 2659 2660 2661 2671
U 11D8  2672 2673 2674 2675 2684 2685 2686 2687
U 11E0  2688 2689 2690 2691 2692 2693 2694 2695
U 11E8  2696 2705 2706 2707 2708 2709 2710 2711
U 11F0  2712 2713 2714 2715 2716 2717 2718 2720
U 11F8  2721 2722 2723 2724 2728 2729 2730 2731
U 1200  2732 2733 2734 2735 2738 2739 2740 2741
U 1208  2743 2744 2745 2746 2747 2753 2754 2755
U 1210  2756 2757 2803 2804 2805 2806 2807 2808
U 1218  2811 2813 2814 2845 2893 2894 2895 2896
U 1220  2897 2898 2899 2900 2901 2904 2905 2906
U 1228  2907 2908 2911 2912 2913 2914 2915 2924
U 1230  2925 2926 2935 2936 2937 2938 2939 2940
U 1238  2949 2950 2951 2952 2953 2986 2987 2988
U 1240  2989 2990 2991 2992 2993 2996 2997 2998
U 1248  2999 3000 3003 3004 3005 3034 3035 3036
U 1250  3037 3038 3041 3042 3043 3044 3045 3048
U 1258  3049 3050 3051 3052 3084 3085 3086 3087
U 1260  3088 3091 3092 3093 3094 3095 3098 3099
U 1268  3100 3101 3102 3133 3134 3135 3136 3137
U 1270  3140 3141 3142 3143 3144 3147 3148 3149
U 1278  3150 3151 3184 3185 3186 3189 3190 3191
U 1280  3192 3193 3194 3195 3198 3199 3200 3201
U 1288  3202 3237 3238 3239 3240 3241 3242 3243
U 1290  3244 3245 3248 3249 3250 3251 3252 3253
U 1298  3254 3255 3256 3257 3258 3259 3260 3296
U 12A0  3302 3303 3304 3305 3306 3307 3308 3309
U 12A8  3310 3311 2113: 3312 3313 3314 3315 3316
U 12B0  3317 3318 3349 3350 3351 3354 3355 3356
U 12B8  3357 3358 3359 3360 3361 3407 3408 3409
U 12C0  3415 3416 3417 3418 3424 3425 3426 3427
U 12C8  3428 3429 3435 3436 3437 3438 3439 3440
U 12D0  3441 3460 3461 3467 3468 3469 3470 3476
U 12D8  3477 3478 3479 3480 3481 3482 3488 3489
U 12E0  3490 3496 3497 3498 3499 3500 3501 3502
U 12E8  3533 3534 3535 3545 3546 3552 3553 3554
U 12F0  3555 3556 3557 3563 3564 3565 3566 3567
U 12F8  3568 3569 3581 3582 3583 3586 3587 3588
U 1300  3594 3595 3596 3597 3598 3599 3605 3606
U 1308  3607 3608 3614 3615 3616 3617 3618 3619
U 1310  3620 3656 3663 3664 3667 3668 3669 3704
U 1318  3705 3706 3707 3708 3709 3710 3711 3712
U 1320  3713 3714 3717 3720 3721 3722 3723 3724
U 1328  3734
U 1330 - 13EF Unused
U 13F0  2036: 2037: 2038: 2039: 2040: 2041: 2042: 2043:
U 13F8  2044: 2045: 2046: 2047: 2048: 2049: 2050: 2051:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR51	825
Used	825
Remaining	

Total microwords used in memory U: 825
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR51.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR52.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 1753

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1850 Micro Trap Handler and LSI-11 Support Routines
; 1851   Micro Trap Handler
; 1931 Micro Monitor Interface Routines
; 1932   Newtest Routine
; 1988   Error Loop Routine
; 2005   Error 1 Routine
; 2030   ERROR1 WITH PARAMETER
; 2051   Error 2 Routine
; 2077   ERROR 2 WITH PARAMETER
; 2096   Micro-Monitor Call
; 2110   Reserved Control Store
; 2137   Set Argument Count
; 2160   Increment Subtest Number
; 2180 Diagnostic Specific Subroutines
; 2181   Select Controller
; 2219   START TEST
; 2428   SELECT LOWER CONTROLLER
; 2476   SELECT UPPER CONTROLLER
; 2524   Set Diagnostic Mode Routine
; 2553   SBI Unjam Routine
; 2603   RESET SUBTEST NUMBER
; 2626   Initialize the SBI
; 2654   Disable Cache Routine
; 2676   Enable Cache
; 2697   Wait Loop Routine
; 2730   Check for Interrupt Routine
; 2777   CHECK UTRAP
; 2809   Set Trap Mask
; 2837   Find MS780-E Memory
; 2978   Generate IO Address
; 3003   Set Starting Address
; 3036   ENABLE NEXT MS780-E
; 3039   ENABLE NEXT MS780-E
; 3091   Set ECC Bits
; 3121   MS780-E/H CNFG REG CLEANUP
; 3191   CHECK SBI.ERR
; 3222   64K OR 256K CHIPS
; 3276   GET MEMORY SIZE
; 3335   TYPE ASCII MESSAGE ROUTINE
; 3376   TYPE ASCII MESSAGE WITH CRLF
; 3419   TYPE NUMERIC ROUTINE
; 3459   TYPE NUMERIC WITH CRLF
; 3501   COMMON CODE FOR TYPE NUMERIC ROUTINES
; 3553   COMMON CODE FOR TYPE ASCII AND NUMERIC
; 3610   DECREMENT TO ROUTINE
; 3651   ASCII MESSAGES
; 3661 TEST 1F7 SINGLE BIT ERROR ON MASKED WRITES
; 4107 TEST 1F8 DOUBLE BIT ERROR ON MASKED WRITES
; 4310 TEST 1F9 QUICK VERIFY ARRAY DATA INTEGRITY TEST

```
; 4808 TEST 1FA RESERVED
; 4813 DUMMY NEWTST
```

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR52
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ; MODIFIED BY:
;1836  ;
;1837  ;  1.1 March 10, 1983  Paul Hakala
;1838  ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839  ;   the correct memory address when 256k ram chips are
;1840  ;   present.
;1841  ;
;1842  ;
;1843  ;
;1844  ;
;1845  ;
;1846  ;*****************************************************************************
;1847  ;
```

```
;1848  .PAGE
;1849
;1850  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1851  .TOC "   Micro Trap Handler"
;1852 ;++
;1853 ; FUNCTIONAL DESCRIPTION:
;1854 ;
;1855 ; This routine is responsible for 'catching' micro-traps
;1856 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1857 ; contains the Trap-Expected Mask. If the specified bit is set in
;1858 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1859 ; an error call is made.
;1860 ;
;1861 ; INPUT PARAMETERS:
;1862 ;
;1863 ; R[0F] - Expected Trap Mask
;1864 ;  Trap Code Function
;1865 ;  ---------      --------
;1866 ;       0  System Init
;1867 ;       1  Data Unaligned
;1868 ;       2  Cross Page
;1869 ;       3  TB Modify
;1870 ;       4  TB Protection
;1871 ;       6  TB Miss
;1872 ;       7  TB Parity
;1873 ;       8  Cache Parity
;1874 ;       9  Reserved Floating Operand
;1875 ;       C  Read Data Substitute
;1876 ;       D  Time out
;1877 ;       F  Control Store Parity Error
;1878 ;      10  Odd Address
;1879 ;
;1890 ; OUTPUT PARAMETERS:
;1881 ;
;1882 ; R[0F] - Mask bit is cleared.
;1883 ;
;1884 ; DATA: Micro PC of Micro Trap
;1885 ;  Trap Code (See Above)
;1886 ;  Fault Status Register
;1887 ;  SBI Error Register
;1888 ;  Timeout Address Register
;1889 ;  Maintenance Register
;1890 ;  Cache Parity Register
;1891 ;  TB Error Register 1
;1892 ;  TB Error Register 0
;1893 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1894 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1895 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1896 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1897 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1898 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1899 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1900 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1901 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1902 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000,003C,8580,0800,0084,7001  ;1903 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000,003C,8980,0800,0084,7001  ;1904 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000,003C,6580,0800,0084,7001  ;1905 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000,003C,6180,0800,0084,7003  ;1906 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1907
U 1001, 0000,003C,81F0,2C00,0000,1036  ;1908 trph0: q_id[ustack]  ; Get upc of trap
U 1036, 0C00,003C,01E0,0800,0000,103B  ;1909  q_d,d_q   ; Setup to put it back on stack
U 103B, 0C00,003C,81E0,3C00,0000,103E  ;1910  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 103E, 0000,003C,01C0,0A78,0000,1043  ;1911  q_r[0f]   ; Get the Expected Trap Mask
        ;1912  alu_q.andnot.mask,
U 1043, 0001,2024,0180,0800,0010,1047  ;1913   clk.ubcc ; Was trap expected?
U 1047, 0000,013C,0180,0800,0000,1002  ;1914  z?
        ;1915 =0 r[0f]_alu,  ; It was, clear the mask and return
        ;1916      alu_q.and.mask, ; ...
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1917      return0  ; ...
U 1003, 0018,0038,1D80,09E8,0000,1034  ;1918 trph1: rc[0d]_sc  ; Output the Trap Code
U 1034, 0000,003D,D980,0800,0084,719A  ;1919 =0 setrcf[.9]  ; Set output count
U 1035, 0000,003C,49F0,2C00,0000,104B  ;1920  q_id[tber0]  ; Output all the error registers
U 104B, 0001,203C,4DF0,2DB0,0000,104F  ;1921  rc[6]_q,q_id[tber1] ; ...
U 104F, 0001,203C,65F0,2DB8,0000,1053  ;1922  rc[7]_q,q_id[sbi.err] ; ...
U 1053, 0001,203C,6DF0,2DD8,0000,1057  ;1923  rc[0b]_q,q_id[fault] ; ...
U 1057, 0001,203C,75F0,2DE0,0000,105B  ;1924  rc[0c]_q,q_id[maint] ; ...
U 105B, 0001,203C,79F0,2DC8,0000,105F  ;1925  rc[9]_q,q_id[parity] ; ...
U 105F, 0001,203C,69F0,2DC0,0000,1109  ;1926  rc[8]_q,q_id[time.addr] ; ...
U 1109, 0001,203C,81F0,2DD0,0000,1000  ;1927  rc[0a]_q,q_id[ustack] ; ...
        ;1928 1000: ERROR1[.C],  ;
U 1000, 0001,203D,8580,09F0,0084,7058  ;1929   rc[0e]_q ; Report the error
```

```
            ;1930  .PAGE
            ;1931  .TOC " Micro Monitor Interface Routines"
            ;1932  .TOC "   Newtest Routine"
            ;1933 ;++
            ;1934 ; FUNCTIONAL DESCRIPTION:
            ;1935 ;
            ;1936 ; This routine Initializes the following registers:
            ;1937 ;  PSL    to 1F
            ;1938 ;  CES    to 0
            ;1939 ;  ACC.CS    to disable accellerator
            ;1940 ;  SIR    to 0
            ;1941 ;  CLK.CS    to stop interval timer
            ;1942 ;  TBER0    to disable the TB
            ;1943 ;  SBI.MAINT to     0
            ;1944 ;  SBI.ERR    to 0
            ;1945 ;  FAULT    to 0
            ;1946 ;  COMP    to 0
            ;1947 ;  RC[0E]    to  0
            ;1948 ;  R[0F]    to 0
            ;1949 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1950 ; A SCOPE call is then made to the Monitor.
            ;1951 ;
            ;1952 ; CALLING CONSTRAINT:
            ;1953 ;
            ;1954 ; =0
            ;1955 ;
            ;1956 ; SIDE EFFECTS:
            ;1957 ;
            ;1958 ; D Reg is destroyed
            ;1959 ; SC is destroyed
            ;1960 ;--
U 1180, 0000,003C,65F8,0800,0084,7181  ;1961 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1181, 0818,0038,8D80,0800,0000,1182  ;1962  d_k[.1f]  ; D=1F
U 1182, 0D00,003C,0180,0800,0000,1183  ;1963  d_dal.sc  ; D=001F0000
U 1183, 0000,003C,3D80,3C00,0000,1184  ;1964  id[psl]_d  ; psl = 001F0000
U 1184, 0818,0038,0580,0800,0000,1185  ;1965  d_k[.1]  ; Clear the CES register
U 1185, 0D00,003C,0180,0800,0000,1186  ;1966  d_dal.sc  ; D = 00010000
U 1186, 0F00,003C,3180,3C00,0000,1187  ;1967  id[ces]_d,d_0  ; ces = 00010000
U 1187, 0000,003C,5D80,3C00,0000,1188  ;1968  id[acc.cs]_d  ; acc.cs = 0
U 1188, 0000,003C,3980,3C00,0000,1189  ;1969  id[sir]_d  ; sir = 0
U 1189, 0000,003C,2980,3C00,0000,118A  ;1970  id[clk.cs]_d  ; clk.cs = 0
U 118A, 0000,003C,4980,3C00,0000,103C  ;1971  id[tber0]_d  ; tber0 = 0
U 103C, 0000,003D,0180,0800,0000,11B6  ;1972 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1973  intrpt.strobe,  ; Strobe interrupts
U 103D, 0818,0038,C180,0800,4000,118B  ;1974      d_k[.ffff]  ; Setup to clear SBI error register and
U 118B, 0801,0028,6580,3C00,0000,118C  ;1975  id[sbi.err]_d,d_not.d ; and fault register
U 118C, 0F00,003C,6D80,3C00,0000,118D  ;1976  id[fault]_d,d_0  ; ...
U 118D, 0000,003C,6D80,3C00,0000,118E  ;1977  id[fault]_d  ; fault = 0
U 118E, 0000,003C,7580,3C00,0000,118F  ;1978  id[maint]_d  ; MAINT = 0
U 118F, 0000,003C,6580,3C00,0000,1190  ;1979  id[sbi.err]_d  ; sbi error reg = 0
U 1190, 0000,003C,7180,3C00,0000,1191  ;1980  id[comp]_d  ; comp reg = 0
U 1191, 0000,003C,E580,3C00,0000,1192  ;1981  ID[39]_d  ; Clear code for subroutine START.TEST
U 1192, 0001,003C,0180,09F0,0000,1193  ;1982  rc[0e]_d  ;
U 1193, 0001,003C,0180,0AF8,0000,1194  ;1983  r[0f]_d  ; Clear expected trap mask
U 1194, 0001,003C,0180,09F8,0000,104C  ;1984  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 104C, 0000,003D,0180,0800,0000,11BC  ;1985 =0 call[disable.cache] ; Disable the cache
U 104D, 0F03,003C,0180,09E8,0000,105E  ;1986  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1987  .PAGE
            ;1988  .TOC "   Error Loop Routine"
            ;1989 ;++
            ;1990 ; FUNCTIONAL DESCRIPTION:
            ;1991 ;
            ;1992 ; This routine is called with the "ERLOOP" macro. The
            ;1993 ; routine calls the Micro Monitor to setup an error loop.
            ;1994 ;
            ;1995 ; CALLING CONSTRAINT:
            ;1996 ;
            ;1997 ; =0
            ;1998 ;
            ;1999 ; SIDE EFFECTS:
            ;2000 ;
            ;2001 ; D Reg - Destroyed
            ;2002 ;--
U 1195, 0818,0038,0980,0800,0000,105E  ;2003 ERLOOP: d_k[.2],j/callcon
```

```
            ;2004  .PAGE
            ;2005  .TOC "   Error 1 Routine"
            ;2006 ;**
            ;2007 ; FUNCTIONAL DESCRIPTION:
            ;2008 ;
            ;2009 ; This routine is used to report an error to the user. This
            ;2010 ; routine is used when you do not wish to continue in the
            ;2011 ; same test after the call to the Monitor.
            ;2012 ;
            ;2013 ; CALLING CONSTRAINT:
            ;2014 ;
            ;2015 ; None
            ;2016 ;
            ;2017 ; INPUTS:
            ;2018 ;
            ;2019 ; rc[0f]<15:8> - Contain the subtest number
            ;2020 ;
            ;2021 ; OUTPUTS:
            ;2022 ;
            ;2023 ; D<15:8> - Subtest number
            ;2024 ; D<7:0> - Error 1 Monitor Code
            ;2025 ;--
U 1196, 0810,0038,0180,0978,0000,1197  ;2026 ERROR1: d_rc[0f]  ; Get the subtest number
U 1197, 0819,0034,4D80,0800,0000,104E  ;2027  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2028 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2029  .PAGE
            ;2030  .TOC "   ERROR1 WITH PARAMETER"
            ;2031 ;++
            ;2032 ; FUNCTIONAL DESCRIPTION:
            ;2033 ;
            ;2034 ; This subroutine takes  as parameter the number of arguments
            ;2035 ; to be printed to the console in the case of an error logout.
            ;2036 ;
            ;2037 ; CALLING CONSTRAINT:
            ;2038 ;
            ;2039 ; =0
            ;2040 ;INPUTS:
            ;2041 ;
            ;2042 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2043 ;--
            ;2044 =0
            ;2045 ERR1.ARG:
U 1058, 0000,003D,0180,0800,0000,119A  ;2046  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1059, 0000,003C,0180,0800,0000,1196  ;2047  J/ERROR1  ; Go to ERROR1
            ;2048 ;
            ;2049 ;
```

```
            ;2050  .PAGE
            ;2051  .TOC "    Error 2 Routine"
            ;2052 ;++
            ;2053 ; FUNCTIONAL DESCRIPTION:
            ;2054 ;
            ;2055 ; This routine is used  to report an error to the user. This
            ;2056 ; routine is used when you wish to continue in the same test
            ;2057 ;       after the call to the Monitor.
            ;2058 ;
            ;2059 ; CALLING CONSTRAINT:
            ;2060 ;
            ;2061 ; =0
            ;2062 ;
            ;2063 ; INPUTS:
            ;2064 ;
            ;2065 ; rc[0f]<15:8> - Contain the subtest number
            ;2066 ;
            ;2067 ; OUTPUTS:
            ;2068 ;
            ;2069 ; D<15:8> - Subtest number
            ;2070 ; D<7:0> - Error 2 Monitor Code
            ;2071 ;--
U 1198, 0810,0038,0180,0978,0000,1199  ;2072 ERROR2: d_rc[0f]  ; Get the subtest number
U 1199, 0819,0034,4D80,0800,0000,105A  ;2073  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2074 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2075 ;
```

```
            ;2076  .PAGE
            ;2077  .TOC "   ERROR 2 WITH PARAMETER"
            ;2078 ;++
            ;2079 ; FUNCTIONAL DESCRIPTION:
            ;2080 ;
            ;2081 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2082 ; except that in this case after setting the number of arguments
            ;2083 ; too the console, control is transferred to routine ERROR2.
            ;2084 ;
            ;2085 ; INPUTS:
            ;2086 ;
            ;2087 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2088 ;
            ;2089 ;--
            ;2090 =0
            ;2091 ERR2.ARG:
U 105C, 0000,003D,0180,0800,0000,119A  ;2092  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 105D, 0000,003C,0180,0800,0000,1198  ;2093  J/ERROR2  ; Go to ERROR2
            ;2094 ;
```

```
            ;2095  .PAGE
            ;2096  .TOC "   Micro-Monitor Call"
            ;2097 ;+
            ;2098 ; FUNCTIONAL DESCRIPTION:
            ;2099 ;
            ;2100 ; This micro word calls the micro monitor.
            ;2101 ;
            ;2102 ; EXPLICIT INPUTS:
            ;2103 ;
            ;2104 ; D reg must contain valid monitor code.
            ;2105 ;--
            ;2106 105E:
            ;2107 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2108  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2109   .PAGE
            ;2110   .TOC "   Reserved Control Store"
            ;2111 ;++
            ;2112 ; FUNCTIONAL DESCRIPTION:
            ;2113 ;
            ;2114 ; The following 16 locations must be reserved so the monitor
            ;2115 ; can use them to perform various functions that require
            ;2116 ; the execution of micro-code.
            ;2117 ;--
U 13F0, 0000,003C,0180,0800,0000,13F1   ;2118 13F0: nop
U 13F1, 0000,003C,0180,0800,0000,13F2   ;2119 13F1: nop
U 13F2, 0000,003C,0180,0800,0000,13F3   ;2120 13F2: nop
U 13F3, 0000,003C,0180,0800,0000,13F4   ;2121 13F3: nop
U 13F4, 0000,003C,0180,0800,0000,13F5   ;2122 13F4: nop
U 13F5, 0000,003C,0180,0800,0000,13F6   ;2123 13F5: nop
U 13F6, 0000,003C,0180,0800,0000,13F7   ;2124 13F6: nop
U 13F7, 0000,003C,0180,0800,0000,13F8   ;2125 13F7: nop
U 13F8, 0000,003C,0180,0800,0000,13F9   ;2126 13F8: nop
U 13F9, 0000,003C,0180,0800,0000,13FA   ;2127 13F9: nop
U 13FA, 0000,003C,0180,0800,0000,13FB   ;2128 13FA: nop
U 13FB, 0000,003C,0180,0800,0000,13FC   ;2129 13FB: nop
U 13FC, 0000,003C,0180,0800,0000,13FD   ;2130 13FC: nop
U 13FD, 0000,003C,0180,0800,0000,13FE   ;2131 13FD: nop
U 13FE, 0000,003C,0180,0800,0000,13FF   ;2132 13FE: nop
U 13FF, 0000,003C,0180,0800,0000,12AA   ;2133 13FF: nop
U 12AA, 0000,003C,0180,0800,0000,119A   ;2134 12AA: nop ; This location is permanently blasted in the FPLA
            ;2135   ; to cause a micro ECO to location 12AA.
```

```
            ;2136  .PAGE
            ;2137  .TOC "   Set Argument Count"
            ;2138 ;++
            ;2139 ; FUNCTIONAL DESCRIPTION:
            ;2140 ;
            ;2141 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2142 ; to the console.
            ;2143 ;
            ;2144 ; CALLING CONSTRAINT:
            ;2145 ;
            ;2146 ; =0
            ;2147 ;
            ;2148 ; INPUTS:
            ;2149 ;
            ;2150 ; SC - Contains new value of argument count
            ;2151 ;
            ;2152 ; SIDE EFFECTS:
            ;2153 ;
            ;2154 ; Q is destroyed
            ;2155 ;--
U 119A, 0010,0038,01C0,0978,0000,119B  ;2156 SETRCF: q_rc[0f]  ; Get the argument count
U 119B, 0019,2034,4DC0,0800,0000,119C  ;2157  q_q.and.k[.ff00] ; ...
U 119C, 0019,2032,1D80,09F8,0000,0001  ;2158  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2159  .PAGE
            ;2160  .TOC "   Increment Subtest Number"
            ;2161 ;++
            ;2162 ; FUNCTIONAL DESCRIPTION:
            ;2163 ;
            ;2164 ; This routine is used to increment the subtest number
            ;2165 ; in RC[0F]<15:8>.
            ;2166 ;
            ;2167 ; CALLING CONSTRAINT:
            ;2168 ;
            ;2169 ; =0
            ;2170 ;
            ;2171 ; SIDE EFFECTS:
            ;2172 ;
            ;2173 ; D register is destroyed
            ;2174 ;--
            ;2175 subtst:
U 119D, 0810,0038,4D80,0978,0000,119E  ;2176  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2177  rc[0f]_d+k[.ff00], ; Increment and return
U 119E, 0019,4016,4D80,09F8,0000,0001  ;2178      word,return1 ; (Subtest number is negative)
```

```
            ;2179  .PAGE
            ;2180  .TOC " Diagnostic Specific Subroutines"
            ;2181  .TOC "   Select Controller"
            ;2182 ;++
            ;2183 ; FUNCTIONAL DESCRIPTION:
            ;2184 ;
            ;2185 ; This routine is used to put the memory system into the
            ;2186 ; specified configuration with respect to controller select.
            ;2187 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2188 ; a RETURN2 is made.
            ;2189 ;
            ;2190 ; CALLING CONSTRAINT:
            ;2191 ;
            ;2192 ; =00
            ;2193 ;
            ;2194 ; INPUTS:
            ;2195 ;
            ;2196 ; d - Contains value to write to bits<2:0> of Register A
            ;2197 ; rc[0e] - Address of Register A
            ;2198 ;
            ;2199 ; SIDE EFFECTS:
            ;2200 ;
            ;2201 ; sc,d,va are destroyed
            ;2202 ;--
            ;2203 SELECT.CNTRLR:
U 119F. 0010.0038.0180.0970.0284.71A0  ;2204  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11A0. 0801.001C.0180.0800.0000.11A1  ;2205  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11A1. 0000.003C.0180.A800.0000.11A2  ;2206  cache.p_d[long]  ; Write the configuration Register
U 11A2. 0818.0038.5D80.0800.0000.11A3  ;2207  d_k[.7]  ; Get Misconfiguration bits
U 11A3. 0000.003C.61F8.0800.0084.71A4  ;2208  sc_k[.F],q_0  ; ...
U 11A4. 0D00.003C.0180.0800.0000.11A5  ;2209  d_dal.sc  ; ... D = x38000
U 11A5. 0000.003C.01E0.0800.0000.11A6  ;2210  q_d  ; Save mask
U 11A6. 0000.003C.0180.C800.0000.11A7  ;2211  d[long]_cache.p  ; Read CNFG A Reg and
            ;2212  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11A7. 001D.2034.0180.0800.0010.11A8  ;2213  clk.ubcc ; ...
U 11A8. 0000.013C.0180.0800.0000.1068  ;2214  z?  ; Branch if no
U 1068. 0000.003E.0180.0800.0000.0001  ;2215 =0 RETURN1  ; Bad configuration
U 1069. 0000.003E.0180.0800.0000.0002  ;2216  RETURN2  ; Configuration ok
            ;2217
```

```
;2218  .PAGE
;2219  .TOC "   START TEST"
;2220 ;*************************************************************
;2221 ;**
;2222 ;
;2223 ; Functional Description:
;2224 ; ---------------------
;2225 ;
;2226 ;         This subroutine will be called by all tests that  check  the
;2227 ;    controllers,  or  those  tests  that  have  to switch between the
;2228 ;    controllers when executing.   Its  main  functions  are:   select
;2229 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2230 ;    test that called it, call out an error  if  neither  one  of  the
;2231 ;    controllers can be configured properly, if all the controllers on
;2232 ;    a MS780-E/H were configured and tested, it should select the next
;2233 ;    MS780-E/H on the SBI and repeat the above sequence.
;2234 ;    A test making use of this  subroutine  should  be  configured  as
;2235 ;    follows:
;2236 ;
;2237 ;
;2238 ;        Begin TEST.XX
;2239 ;         :
;2240 ;         :   Call NEWTST
;2241 ;         :   Call FIND.MS780E
;2242 ;         :   IF No MS780-E's on the SBI
;2243 ;         :     THEN
;2244 ;         :         Skip to End of TEST.XX
;2245 ;         :     ELSE
;2246 ;         :
;2247 ;         : RESTART.TEST.XX:
;2248 ;         :
;2249 ;         :   Call START TEST
;2250 ;         :   IF Return1 from START.TEST
;2251 ;         :     THEN
;2252 ;         :         Skip to End of TEST.XX
;2253 ;         :     ELSE.
;2254 ;         :
;2255 ;         ~
;2256 ;         ~
;2257 ;         :
;2258 ;         :   Body of TEST.XX
;2259 ;         :
;2260 ;         ~
;2261 ;         ~
;2262 ;         :   Jump RESTART.TEST.XX
;2263 ;         :
;2264 ;        End TEST.XX
;2265 ;
;2266 ;
;2267 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2268 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2269 ;    should be passed when the subroutine is called a second, third or
;2270 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2271 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2272 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2273 ;    Register 39 can be interpreted as follows:
;2274 ;
;2275 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2276 ;               called the first time.  When this code is encountered
;2277 ;               this subroutine will try to select the lower controller.
;2278 ;               If the lower controller is misconfigured, the pointer in
;2279 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2280 ;               is re-entered.
;2281 ;               If, however there is no misconfiguration, the value in
;2282 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2283 ;               calling test.
;2284 ;
;2285 ;     b01   -   This value signifies that an attempt at configuring the
;2286 ;               lower controller failed, and the subroutine will attempt
;2287 ;               to select the upper controller.
;2288 ;               If the upper controller is also misconfigured execution
;2289 ;               stops with a call to ERROR1.
;2290 ;               If there is no misconfiguration on this controller, then
;2291 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2292 ;               made to the calling test.
;2293 ;
;2294 ;     b10   -   This value signifies that the lower controller was selected
;2295 ;               previously and the test was executed once.  If the
;2296 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2297 ;               39 is loaded with b11 and an attempt is made to select the
;2298 ;               upper controller.
;2299 ;               If there is a misconfiguration on this controller, this
;2300 ;               subroutine is just re-entered.
;2301 ;               Otherwise, a Return2 will be made to the calling test.
;2302 ;
;2303 ;     b11   -   This code identifies the cases in which the test has
;2304 ;               been executed at least once (i.e., at least one of the
;2305 ;               controllers on the MS780-E unit under test could be
;2306 ;               configured properly).  When this code is detected the
;2307 ;               subroutine clears ID Reg 39 and makes a call to
;2308 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2309 ;               tested then the subroutine executes a Return1.
;2310 ;               Otherwise the subroutine is re-entered.
;2311 ;
;2312 ;
;2313 ; Calling Constraint: =00
;2314 ; -------------------
;2315 ;
;2316 ;
;2317 ; Inputs:
;2318 ; ------
;2319 ;
;2320 ; ID Register 39 must be cleared by NEWTST
;2321 ;
;2322 ;
;2323 ; Outputs:
;2324 ; -------
;2325 ;
;2326 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2327 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
            ;2328 ;         to be tested
            ;2329 ;
            ;2330 ; Side Effects:
            ;2331 ; ------------
            ;2332 ;
            ;2333 ; Contents of the following regiters will be destroyed:
            ;2334 ;
            ;2335 ; D, Q
            ;2336 ;
            ;2337 ;--
            ;2338 ;*************************************************************
            ;2339 =0
            ;2340 START.TEST:
U 106C, 0000,003D,0180,0800,0000,11B7  ;2341  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 106D, 0000,003C,0180,0800,0000,107C  ;2342  NOP    ; ...tests that have more than one
            ;2343 =0     ; ...subtest.)
U 107C, 0000,003D,0180,0800,0000,1195  ;2344  ERLOOP    ; Set up the error loop for the
            ;2345       ; ...eventuality that the MS780-E/H
            ;2346       ; ...currently under test has no
            ;2347       ; ...Controllers
            ;2348 RE.ENTRY:    ; Re entry point for this routine
U 107D, 0000,003C,E5F0,2C00,0000,11A9  ;2349  Q_ID[39]    ; Get the code
U 11A9, 0C00,003C,0180,0800,0000,11AA  ;2350  D_Q    ; Put it in the D Register
U 11AA, 0000,193C,0180,0800,0000,100C  ;2351  DI-0?    ; Branch on the code
U 100C, 0000,003C,0180,0800,0000,1008  ;2352 =1100 J/STRT.CASE00    ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1010  ;2353  J/STRT.CASE01    ; Code is 01
U 100E, 0000,003C,0180,0800,0000,11AF  ;2354  J/STRT.CASE10    ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1017  ;2355  J/STRT.CASE11    ; Code is 11
            ;2356 ;
            ;2357 ;
            ;2358 ; At this point the code in ID[39] was 00
            ;2359 ;
            ;2360 ;
            ;2361 =00
            ;2362 STRT.CASE00:
U 1008, 0000,003D,0180,0800,0000,1024  ;2363  CALL[SELECT.LOWER]  ; Try to select the lower Controller
            ;2364  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1009, 0818,0038,0580,0800,0000,11AB  ;2365   D_K[.1]    ; Set up the code for re entry to this
            ;2366       ; ...routine ( 01 )
U 100A, 0818,0038,0980,0800,0000,100B  ;2367  D_K[.2]    ; Set up the code for a next call to
            ;2368       ; ...START.TEST indicating that the
            ;2369       ; ...Lower Controller was configured
            ;2370       ; ...( 10 )
            ;2371  ID[39]_D,    ; Save the code in ID[39]
U 100B, 0000,003E,E580,3C00,0000,0002  ;2372   RETURN2    ; Let the test know that the Lower
            ;2373       ; ...has been configured
            ;2374 STRT.LOW.MIS:
            ;2375  ID[39]_D,    ; Save code in ID[39] signifing that
U 11AB, 0000,003C,E580,3C00,0000,107D  ;2376   J/RE.ENTRY  ; ...the Lower Controller was Misconfig
            ;2377       ; ...and re enter this subroutine
            ;2378
            ;2379 ;
            ;2380 ;
            ;2381 ; At this point the code is 01
            ;2382 ;
```

```
            ;2383 ;
            ;2384 =00
            ;2385 STRT.CASE01:
U 1010, 0000,003D,0180,0800,0000,1028  ;2386  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2387  J/STRT.UPR.MIS,   ; Upper Controller is Misconfigured
U 1011, 0F00,003C,0180,0800,0000,11AC  ;2388   D_0   ; Prepare to clear the code
U 1012, 0818,0038,0D80,0800,0000,1013  ;2389  D_K[.3]    ; Upper controller was configured
            ;2390       ; ...thus the code must be changed
            ;2391       ; ...accordingly ( 11 )
            ;2392  ID[39]_D,   ; Save the code
U 1013, 0000,003E,E580,3C00,0000,0002  ;2393   RETURN2   ; Let the test know that the Upper
            ;2394       ; ...Controller has been configured
            ;2395 STRT.UPR.MIS:
U 11AC, 0000,003C,E580,3C00,0000,11AD  ;2396  ID[39]_D   ; Save the Code ( 00 )
U 11AD, 0018,0038,0D80,09E8,0000,11AE  ;2397  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 11AE, 0000,003D,0980,0800,0084,7058  ;2398  ERROR1[.2]   ; Flag the error.
            ;2399 ;
            ;2400 ;
            ;2401 ; At this point the code is 10
            ;2402 ;
            ;2403 ;
            ;2404 STRT.CASE10:
U 11AF, 0818,0038,0D80,0800,0000,1014  ;2405  D_K[.3]    ; Set the code to 11
            ;2406
            ;2407
            ;2408 =00 ID[39]_D,   ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1028  ;2409   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1015, 0000,003C,0180,0800,0000,107D  ;2410  J/RE.ENTRY   ; Misconfigured Upper Controller
U 1016, 0000,003E,0180,0800,0000,0002  ;2411  RETURN2    ; Upper Controller configured
            ;2412       ; ...let the test know it
            ;2413 ;
            ;2414 ;
            ;2415 ; At this point the code is 11
            ;2416 ;
            ;2417 ;
            ;2418 STRT.CASE11:
U 1017, 0F00,003C,0180,0800,0000,1020  ;2419  D_0    ; Clear the code
            ;2420 =00 ID[39]_D,   ; Save the code
U 1020, 0000,003D,E580,3C00,0000,120D  ;2421   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2422       ; ...on the SBI
U 1021, 0000,003C,0180,0800,0000,107D  ;2423  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2424       ; ...attempt to configure the cntrllrs
U 1022, 0000,003E,0180,0800,0000,0001  ;2425  RETURN1    ; No more MS780-E/Hs found
U 1023, 0000,003C,0180,0800,0000,1024  ;2426  NOP
```

```
;2427  .PAGE
;2428  .TOC "   SELECT LOWER CONTROLLER"
;2429 ;*************************************************************
;2430 ;++
;2431 ;
;2432 ; Functional Description:
;2433 ; ----------------------
;2434 ;
;2435 ; This subroutine can be called to select the lower
;2436 ; Controller on a MS780-E/H.
;2437 ; If the Lower controller is misconfigured then a
;2438 ; RETURN1 will be made.  Otherwise a RETURN2
;2439 ;
;2440 ; Calling Constraint: =00
;2441 ; ------------------
;2442 ;
;2443 ;
;2444 ; Inputs:
;2445 ; ------
;2446 ;
;2447 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2448 ;
;2449 ; Outputs:
;2450 ; -------
;2451 ;
;2452 ; RC[0D] will have the address of CNFG Register C
;2453 ;        ( 2000x008 )
;2454 ;
;2455 ; Side Effects:
;2456 ; ------------
;2457 ;
;2458 ; Contents of the following registers will lost:
;2459 ;
;2460 ; D
;2461 ;
;2462 ; The Lower controller on the MS780-E/H will be configured
;2463 ; when the subroutine is exitted with a RETURN2
;2464 ;--
;2465 ;*************************************************************
;2466 =00
;2467 SELECT.LOWER:
;2468  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1024, 0818,0039,1980,0800,0000,119F  ;2469   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1025, 0000,003E,0180,0800,0000,0001  ;2470  RETURN1    ; Lower Controller Misconfigured
U 1026, 0810,0038,0180,0970,0000,1027  ;2471  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2472  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1027, 0019,0016,0180,09E8,0000,0002  ;2473   RETURN2   ; Let caller know that the controller
;2474       ; ...was configured properly
```

```
;2475  .PAGE
;2476  .TOC "   SELECT UPPER CONTROLLER"
;2477 ;*************************************************************
;2478 ;++
;2479 ;
;2480 ; Functional Description:
;2481 ; ----------------------
;2482 ;
;2483 ; This subroutine can be called to select the upper
;2484 ; Controller on a MS780-E/H.
;2485 ; If the Upper controller is misconfigured then a
;2486 ; RETURN1 will be made.  Otherwise a RETURN2
;2487 ;
;2488 ; Calling Constraint: =00
;2489 ; ------------------
;2490 ;
;2491 ;
;2492 ; Inputs:
;2493 ; ------
;2494 ;
;2495 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2496 ;
;2497 ; Outputs:
;2498 ; -------
;2499 ;
;2500 ; RC[0D] will have the address of CNFG Register D
;2501 ;        ( 2000x010 )
;2502 ;
;2503 ; Side Effects:
;2504 ; ------------
;2505 ;
;2506 ; Contents of the following registers will lost:
;2507 ;
;2508 ; D
;2509 ;
;2510 ; The Upper controller on the MS780-E/H will be configured
;2511 ; when the subroutine is exitted with a RETURN2
;2512 ;--
;2513 ;*************************************************************
;2514 =00
;2515 SELECT.UPPER:
;2516  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,119F  ;2517   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2518  RETURN1    ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2519  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2520  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2521   RETURN2   ; Let caller know that the controller
;2522       ; ...was configured properly
```

```
            ;2523  .PAGE
            ;2524  .TOC "   Set Diagnostic Mode Routine"
            ;2525 ;**
            ;2526 ; FUNCTIONAL DESCRIPTION:
            ;2527 ;
            ;2528 ; This routine is used to set the specified diagnostic mode
            ;2529 ; in Register B. Other bits in the register remain unchanged.
            ;2530 ;
            ;2531 ; CALLING CONSTRAINT:
            ;2532 ;
            ;2533 ; =0
            ;2534 ;
            ;2535 ; INPUTS:
            ;2536 ;
            ;2537 ; d - Contains the mode to set in bits<1:0>
            ;2538 ; rc[0e] - Contains base address of controller
            ;2539 ;
            ;2540 ; SIDE EFFECTS:
            ;2541 ;
            ;2542 ; d,va,sc are destroyed
            ;2543 ;--
            ;2544 SET_DIAG_MODE:
U 11B0, 0010,0038,D9F8,0970,0284,71B1  ;2545  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11B1, 0D00,003C,0180,0803,0000,11B2  ;2546  va_va+4,d_dal.sc ; Shift specified mode into position
U 11B2, 0000,003C,01E0,0800,0000,11B3  ;2547  q_d    ; Save the diagnostic mode bits
U 11B3, 0000,003C,0180,C800,0000,11B4  ;2548  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11B4, 081D,0030,0180,0800,0000,11B5  ;2549  d_d.or.q  ; Set the diagnostic mode bits
U 11B5, 0000,003E,0180,A800,0000,0001  ;2550  cache.p_d[long],return1 ; Set the specified mode and return
            ;2551
```

```
            ;2552  .PAGE
            ;2553  .TOC "    SBI Unjam Routine"
            ;2554 ;**
            ;2555 ; FUNCTIONAL DESCRIPTION:
            ;2556 ;
            ;2557 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2558 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2559 ; and finally HOLD for the last 16 cycles.
            ;2560 ;
            ;2561 ; CALLING CONSTRAINT:
            ;2562 ;
            ;2563 ; =0
            ;2564 ;
            ;2565 ; SIDE EFFECTS:
            ;2566 ;
            ;2567 ; D Reg destroyed
            ;2568 ;--
            ;2569    ;
            ;2570    ; assert hold for 16 cycles
            ;2571    ;
            ;2572 SBI.UNJAM:
            ;2573  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11B6, 0818,0038,6580,8000,0010,107E  ;2574       clk.ubcc  ; set micro cc's for next cycle
            ;2575 =0
            ;2576 SBI.UNJAM.1:
            ;2577  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2578       clk.ubcc,z?, ; ...
U 107E, 0819,0100,0580,8000,0010,107E  ;2579       j/sbi.unjam.1 ; test for completion
            ;2580    ;
            ;2581    ; assert hold and unjam for 16 cycles
            ;2582    ;
            ;2583  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 107F, 0818,0038,6580,8800,0010,1080  ;2584       d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2585 =0
            ;2586 SBI.UNJAM.2:
            ;2587  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2588       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1080, 0819,0100,0580,8800,0010,1080  ;2589       z?,j/sbi.unjam.2 ; ...
            ;2590    ;
            ;2591    ; assert hold for 16 cycles
            ;2592    ;
            ;2593  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1081, 0818,0038,6580,8000,0010,1082  ;2594       clk.ubcc  ; and set cc's for next cycle
            ;2595 =0
            ;2596 SBI.UNJAM.3:
            ;2597  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2598       clk.ubcc,z?, ; ...
U 1082, 0819,0100,0580,8000,0010,1082  ;2599       j/sbi.unjam.3 ; ...
U 1083, 0000,003E,0180,0800,0000,0001  ;2600  return1   ; exit
            ;2601
```

```
            ;2602  .PAGE
            ;2603  .TOC "   RESET SUBTEST NUMBER"
            ;2604 ;**
            ;2605 ; FUNCTIONAL DESCRIPTION:
            ;2606 ;
            ;2607 ; Since some  of the tests will have to be restarted when two
            ;2608 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2609 ; the subtest counter to zero ( only for thoes tests that have
            ;2610 ; more than one subtest). This subroutine may also be necessary
            ;2611 ; when a test is being loop on.
            ;2612 ;
            ;2613 ; CALLING CONSTRAINT:
            ;2614 ;
            ;2615 ; =0
            ;2616 ; SIDE EFFECTS:
            ;2617 ;
            ;2618 ; D is destroyed
            ;2619 ;
            ;2620 ;--
            ;2621 RESET.SUBTST:
            ;2622  RC[0F]_0,   ; Reset subtest number
U 11B7, 0003,003E,0180,09F8,0000,0001  ;2623   RETURN1   ; ...and return
            ;2624
```

```
            ;2625  .PAGE
            ;2626  .TOC "   Initialize the SBI"
            ;2627 ;++
            ;2628 ; FUNCTIONAL DESCRIPTION:
            ;2629 ;
            ;2630 ; This routine performs the following functions:
            ;2631 ;
            ;2632 ;  Executes an SBI Unjam
            ;2633 ;  Clears the SBI Fault Latch
            ;2634 ;  Clears the SBI Error Register
            ;2635 ;
            ;2636 ; CALLING CONSTRAINT:
            ;2637 ;
            ;2638 ; =0
            ;2639 ;
            ;2640 ; SIDE EFFECTS:
            ;2641 ;
            ;2642 ; D & Q Register destroyed
            ;2643 ;--
            ;2644 =0
            ;2645 SBI.INIT:
U 1084, 0000.003D.0180.0800.0000.11B6  ;2646  call[sbi.unjam]  ; Unjam the SBI
U 1085, 0818.0038.C180.0800.0000.11B8  ;2647  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11B8, 0801.0028.6580.3C00.0000.11B9  ;2648  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11B9, 0F00.003C.6D80.3C00.0000.11BA  ;2649  id[fault]_d,d_0
U 11BA, 0000.003C.6D80.3C00.0000.11BB  ;2650  id[fault]_d
U 11BB, 0000.003E.6580.3C00.0000.0001  ;2651  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2652
```

```
            ;2653  .PAGE
            ;2654  .TOC "   Disable Cache Routine"
            ;2655 ;++
            ;2656 ; FUNCTIONAL DESCRIPTION:
            ;2657 ;
            ;2658 ; This routine disables cache hits by setting bits <16:15>
            ;2659 ; in the maintenance register.
            ;2660 ;
            ;2661 ; CALLING SEQUENCE:
            ;2662 ;
            ;2663 ; =0
            ;2664 ;
            ;2665 ; SIDE EFFECTS:
            ;2666 ;
            ;2667 ; D & Q Registers destroyed
            ;2668 ;--
            ;2669 DISABLE.CACHE:
U 11BC, 0818.0038.4580.0800.0000.11BD  ;2670  d_k[.8000]  ; ... and seed constant
U 11BD, 0500.003C.0180.0800.0000.11BE  ;2671  d_d.left  ; Generate final constant
U 11BE, 0819.0030.4580.0800.0000.11BF  ;2672  d_d.or.k[.8000]  ; ... = 00018000
U 11BF, 0000.003E.7580.3C00.0000.0001  ;2673  id[maint]_d,return1 ; Disable cache and return
            ;2674
```

```
            ;2675  .PAGE
            ;2676  .TOC "   Enable Cache"
            ;2677 ;++
            ;2678 ; FUNCTIONAL DESCRIPTION:
            ;2679 ;
            ;2680 ; This routine enables cache group 0 by clearing the force
            ;2681 ; miss bit in the maintenance register.
            ;2682 ;
            ;2683 ; CALLING CONSTRAINT:
            ;2684 ;
            ;2685 ; =0
            ;2686 ;
            ;2687 ; SIDE EFFECTS:
            ;2688 ;
            ;2689 ; D, Q AND SC Registers destroyed
            ;2690 ;--
            ;2691 ENABLE.CACHE:
U 11C0, 0000,003C,75F0,2C00,0000,11C1  ;2692  q_id[maint]  ; Get current maintenance register
U 11C1, 0000,003C,6580,0800,0084,71C2  ;2693  sc_k[.10]  ; Setup to clear bit 16
U 11C2, 0801,2034,0180,0800,0000,11C3  ;2694  d_q.and.mask  ; Clear bit 16
U 11C3, 0000,003E,7580,3C00,0000,0001  ;2695  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2696  .PAGE
            ;2697  .TOC "    Wait Loop Routine"
            ;2698 ;++
            ;2699 ; FUNCTIONAL DESCRIPTION:
            ;2700 ;
            ;2701 ; As the name implies,  this subroutine is used to set up a wait
            ;2702 ; loop for a specified  number of cycles.  This may be necessary
            ;2703 ; when the micro-program has to wait for another event to finish.
            ;2704 ; When the subroutine is entered Register D should be holding the
            ;2705 ; desired numbered of cycles.
            ;2706 ;
            ;2707 ; CALLING CONSTRAINT:
            ;2708 ;
            ;2709 ; =0
            ;2710 ;
            ;2711 ; INPUTS:
            ;2712 ;
            ;2713 ; d - Contains number of cycles to wait
            ;2714 ; alu.z condition code must not be set
            ;2715 ;
            ;2716 ; SIDE EFFECTS:
            ;2717 ;
            ;2718 ; d is destroyed
            ;2719 ;--
            ;2720 WAIT.LOOP:
U 11C4, 0018,0038,6180,0800,0010,1086  ;2721  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2722      ; ...is not set
            ;2723 =0
            ;2724 W.LOOP:
            ;2725  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1086, 0819,0100,0580,0800,0010,1086  ;2726       j/W.LOOP
U 1087, 0000,003E,0180,0800,0000,0001  ;2727  return1    ; exit
            ;2728
```

```
            :2729  .PAGE
            :2730  .TOC "   Check for Interrupt Routine"
            :2731 ;**
            :2732 ; FUNCTIONAL DESCRIPTION:
            :2733 ;
            :2734 ; This routine is used to check for any interrupts at level 16 or higher.
            :2735 ; If an interrupt is active, the following registers are setup:
            :2736 ;  RC[8] - Gets the VECTOR register
            :2737 ;  RC[7] - Gets the SBI ERROR register
            :2738 ;  RC[6] - Gets the FAULT register
            :2739 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            :2740 ; and a RETURN2 is made. Otherwise, a return1 is made.
            :2741 ;
            :2742 ; CALLING CONSTRAINT:
            :2743 ;
            :2744 ; =00
            :2745 ;
            :2746 ; SIDE EFFECTS:
            :2747 ;
            :2748 ; D & Q are destroyed
            :2749 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            :2750 ;--
            :2751 CHECK_INTERRUPT:
            :2752   ;
            :2753   ; First, enable the fault and RDS/CRD interrupts
            :2754   ;
U 11C5, 0818,0038,4580,0800,0000,11C6  ;2755  d_k[.8000]  ; Get data to set interrupt enable
            :2756  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 11C6, 0500,003C,6580,3C00,0000,11C7  ;2757      d_d.left
U 11C7, 0100,003C,0180,0800,0000,11C8  ;2758  d_d.left2  ; D = 40000
U 11C8, 0000,003C,6D80,3C00,0000,11C9  ;2759  id[fault]_d  ; Set fault interrupt enable
            :2760  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 11C9, 0F03,003C,0180,09A8,4000,11CA  ;2761      rc[5]_0  ; and clear interrupt occurred flag
U 11CA, 0000,003C,3D80,3C00,0000,11CB  ;2762  id[psl]_d  ; Clear the PSL
U 11CB, 0000,003C,6580,3C00,0000,11CC  ;2763  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 11CC, 0000,003C,6D80,3C00,0C00,11CD  ;2764  id[fault]_d  ; Clear FAULT Interrupt Enable
U 11CD, 0000,0E3C,0180,0800,0000,1004  ;2765  int?   ; Branch if external interrupt is active
U 1004, 0000,003E,0180,0800,0000,0001  ;2766 =100 return1   ; Exit with no interrupt
U 1005, 0000,003C,0180,0800,0000,1007  ;2767  j/check.int.1  ; Interrupt
U 1006, 0000,003E,0180,0800,0000,0001  ;2768  return1   ; No interrupt
            :2769 =111
            :2770 CHECK.INT.1:
U 1007, 0000,003C,35F0,2C00,0000,11CE  ;2771  q_id[vector]  ; Get ID Vector Register
U 11CE, 0001,203C,65F0,2DC0,0000,11CF  ;2772  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 11CF, 0001,203C,6DF0,2DB8,0000,11D0  ;2773  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 11D0, 0001,203C,0180,09B0,0000,11D1  ;2774  rc[6]_q   ; Save falut reg
U 11D1, 0018,003A,0580,09A8,0000,0002  ;2775  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2776  .PAGE
            ;2777  .TOC "    CHECK UTRAP"
            ;2778 ;++
            ;2779 ;FUNCTIONAL DESCRIPTION:
            ;2780 ;
            ;2781 ; This subroutine is used to check wether a Utrap occured.
            ;2782 ; It takes the contents of the Save Utrap flags register
            ;2783 ; R[0E], and compares them to the contrnts of the Utrap
            ;2784 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2785 ; according to the results of the test  (i.e., contents of
            ;2786 ; these registers are equal or not).
            ;2787 ; CALLING CONSTRAINT:
            ;2788 ;
            ;2789 ; =00
            ;2790 ;
            ;2791 ; INPUTS:
            ;2792 ;
            ;2793 ; R[0E]- Saved Utrap Mask
            ;2794 ; R[0F]- Expected Utrap Mask
            ;2795 ;
            ;2796 ;--
            ;2797 CHECK_UTRAP:
U 11D2, 0800,003C,0180,0A70,0000,11D3  ;2798  D_R[0E]    ; Reg D is loaded with the trap mask
U 11D3, 0001,003C,0180,09C0,0000,11D4  ;2799  RC[08]_D   ; Save expected Utrap flag for error logout
U 11D4, 0800,003C,0180,0A78,0000,11D5  ;2800  D_R[0F]    ; Get recieved Utrap flag to D reg
U 11D5, 0001,003C,0180,09B8,0000,11D6  ;2801  RC[07]_D   ; Save in RC[07]
            ;2802  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 11D6, 000D,0020,0180,0A70,0010,11D7  ;2803          CLK.UBCC
U 11D7, 0003,013C,0180,0AF8,0000,1088  ;2804  Z?,R[0F]_0   ; Branch if no traps
U 1088, 0000,003E,0180,0800,0000,0002  ;2805 =0 RETURN2    ; Utrap occured
U 1089, 0000,003E,0180,0800,0000,0001  ;2806  RETURN1    ; No Utrap  return
               ;2807 ;
```

```
            ;2808  .PAGE
            ;2809  .TOC "   Set Trap Mask"
            ;2810 ;**
            ;2811 ; FUNCTIONAL DESCRIPTION:
            ;2812 ;
            ;2813 ; This routine is used to set a bit in the Micro Trap Mask
            ;2814 ; R[0F].
            ;2815 ;
            ;2816 ; CALLING CONSTRAINT:
            ;2817 ;
            ;2818 ; =0
            ;2819 ;
            ;2820 ; INPUTS:
            ;2821 ;
            ;2822 ; SC - Contains bit number to set.
            ;2823 ;
            ;2824 ; OUTPUTS:
            ;2825 ;
            ;2826 ; rc[0E] - Contains the current trap mask
            ;2827 ;
            ;2828 ; SIDE EFFECTS:
            ;2829 ;
            ;2830 ; d regsiter is destroyed
            ;2831 ;--
            ;2832 SET_TRAP_MASK:
U 11D8, 0800,003C,0180,0A78,0000,11D9  ;2833  d_r[0f]
U 11D9, 0801,001C,0180,0AF8,0000,11DA  ;2834  r[0f]_d.ornot.mask,d_alu ; Set mask
U 11DA, 0001,003E,0180,0AF0,0000,0001  ;2835  r[0E]_d,return1    ; Save mask and return
```

```
            ;2836  .PAGE
            ;2837  .TOC "   Find MS780-E Memory"
            ;2838 ;**
            ;2839 ; FUNCTIONAL DESCRIPTION:
            ;2840 ;
            ;2841 ;          The diagnostic is designed to handle up to 4 (four)
            ;2842 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2843 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2844 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2845 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2846 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2847 ;     first MS780-E found will have its starting address set to 0
            ;2848 ;     (zero).
            ;2849 ;     All the other MS780-E/H's found will have their starting address
            ;2850 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2851 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2852 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2853 ;     current MS780-E unit under test.
            ;2854 ;     If any of: MS780-A, MS780-C or MA780 is found, its address is
            ;2855 ;     set to maximum.
            ;2856 ;
            ;2857 ; CALLING CONSTRAINT:
            ;2858 ;
            ;2859 ; =00
            ;2860 ;
            ;2861 ; OUTPUTS:
            ;2862 ;
            ;2863 ; rc[0e] - Contains address of Configuration Register
            ;2864 ; r[0]  - Contains TR level
            ;2865 ;
            ;2866 ; SIDE EFFECTS:
            ;2867 ;
            ;2868 ; D register is destroyed.
            ;2869 ;--
            ;2870 =0
            ;2871 FIND.MS780E:
U 108A, 0000,003D,0180,0800,0000,1084  ;2872  call[sbi.init]  ; Make sure SBI is enabled
U 108B, 0003,003C,0180,0988,0000,11DB  ;2873  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11DB, 0818,0038,1980,0800,0000,11DC  ;2874  d_k[zero]  ; Clear MS780-E/H counter
U 11DC, 0000,003C,F980,3C00,0000,11DD  ;2875  id[3e]_d  ; ...
U 11DD, 0018,0038,0580,0A80,0000,11DE  ;2876  r[0]_k[.1]  ; Init TR counter
U 11DE, 0F03,003C,0180,09F0,0000,108C  ;2877  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2878     ; ...CNFG A Reg
            ;2879 =0
            ;2880 FIND.MEM.LOOP:
U 108C, 0800,003D,0180,0A00,0000,1204  ;2881  d_r[0],call[generate.io]; Generate address of TR level
U 108D, 0001,003C,0180,09F0,0200,108E  ;2882  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 108E, 0000,003D,8980,0800,0084,71D8  ;2883 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2884  d[long]_cache.p, ; Read the specified tr level
U 108F, 0000,003C,0180,C970,0000,11DF  ;2885   lc_rc[0e]  ; ...
U 11DF, 0000,003C,0180,0A78,0010,11E0  ;2886  alu_r[0f],clk.ubcc ; Check for no response
U 11E0, 0001,013C,0180,0880,0082,1090  ;2887  z?,sc_d,la_ra[0] ; Branch if no adapter here
U 1090, 0000,003C,0180,0800,0000,11E1  ;2888 =0 j/check.adpt.code ; Check for a memory
U 1091, 0000,003C,0180,0800,0000,1200  ;2889  j/find.mem.lp1  ; Try next tr
            ;2890 CHECK.ADPT.CODE:
```

```
U 11E1, 0000,003C,1D80,0800,1404,71E2  ;2891  state_sc.vie.kmx ; Setup to branch on bits<6:4>
U 11E2, 0000,163C,0180,0800,0000,1018  ;2892  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,71E3  ;2893 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,71E4  ;2894  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,1200  ;2895  j/find.mem.lp1  ; Not a memory
U 101B, 0000,003C,0180,0800,0000,1200  ;2896  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,71E9  ;2897  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,1200  ;2898  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,11ED  ;2899  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,11ED  ;2900  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2901 ;
            ;2902 ; Found an MS780-A or -C. Set the starting address
            ;2903 ; to maximum.
            ;2904 ;
U 11E3, 0818,0038,5D80,0800,0000,11E5  ;2905 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11E4, 0818,0038,0580,0800,0000,11E5  ;2906 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2907 MS780.COMMON:
U 11E5, 0819,0030,7180,0800,0000,11E6  ;2908  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11E6, 0000,003C,01F8,0803,0080,D1E7  ;2909  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11E7, 0D00,003C,0180,0800,0000,11E8  ;2910  d_dal.sc  ; Put data in bits<29:14>
            ;2911  cache.p_d[long], ; Set the starting address to maximum
U 11E8, 0000,003C,0180,A800,0000,1200  ;2912       j/find.mem.lp1 ; and continue TR scan
            ;2913 ;
            ;2914 ; Found an MA780. Set the starting address to maximum
            ;2915 ;
            ;2916 MA780.MAX:
U 11E9, 0818,0038,39F8,0803,0000,11EA  ;2917  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11EA, 0D00,003C,0180,0803,0000,11EB  ;2918  d_dal.sc,va_va+4 ; Put in proper position
U 11EB, 0000,003C,0180,0803,0000,11EC  ;2919  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2920  cache.p_d[long], ; Set starting address to maximum
U 11EC, 0000,003C,0180,A800,0000,1200  ;2921       j/find.mem.lp1
            ;2922 ;
            ;2923 ; Found an MS780E
            ;2924 ;
            ;2925 FOUND.MS780E:
U 11ED, 0000,003C,F9F0,2C00,0000,11EE  ;2926  q_id[3e]  ; Set up to see how many MS780-E's
            ;2927  alu_q.xor.k[.3], ; Are there Maximum units?
U 11EE, 0019,2020,0D80,0800,0010,11EF  ;2928   clk.ubcc ; Check condition codes
U 11EF, 0000,013C,0180,0800,0000,1092  ;2929  z?   ; Are we at maximum units for system
U 1092, 0000,003C,0180,0800,0000,11F0  ;2930 =0 J/ms780e.tst  ; No go find how many units ther are
U 1093, 0818,0039,C180,0800,0000,1094  ;2931  d_k[.ffff]  ; Yes set address to maximum
U 1094, 0000,003D,0180,0800,0000,1208  ;2932 =0 call[set.start.addr] ; .....
U 1095, 0000,003C,0180,0800,0000,1200  ;2933  j/find.mem.lp1  ; Go check to see if we are done
            ;2934 ;
            ;2935 MS780E.TST:
            ;2936  alu_rc[01],  ; Check "Found MS780E Flag"
U 11F0, 0010,0038,0180,0908,0010,11F1  ;2937   clk.ubcc ; Check condition codes
U 11F1, 0810,0138,0180,0970,0000,1096  ;2938  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2939 =0 j/not.first.ms780e, ; No
U 1096, 0818,0038,C180,0800,0000,109A  ;2940   d_k[.ffff] ; Set starting address
U 1097, 0001,003C,0180,0988,0000,11F2  ;2941  rc[01]_d  ; Yes put base address in rc[01]
U 11F2, 0818,0038,7980,0800,0000,11F3  ;2942  d_k[.30]  ; Set up SC to point to first unit
U 11F3, 0019,0014,F580,0800,0082,11F4  ;2943  alu_d+k[.a],sc_alu ; ...
U 11F4, 0810,0038,0180,0908,0000,11F5  ;2944  d_rc[01]  ; Put base address of unit in D
U 11F5, 0000,003C,0180,3400,0000,11F6  ;2945  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 11F6, 0F00,003C,0180,0800,0000,1098  ;2946  d_0   ; Prepare to set starting address to Zero
U 1098, 0000,003D,0180,0800,0000,1208  ;2947 =0 call[set.start.addr] ; Go do it
        ;2948  j/find.mem.lp1,  ; Check to see if we are done
U 1099, 0003,003C,0180,09E8,0000,1200  ;2949   rc[0d]_0 ; ....
        ;2950 =0
        ;2951 NOT.FIRST.MS780E:
U 109A, 0000,003D,0180,0800,0000,1208  ;2952  call[set.start.addr] ; Go set starting address to maximum
U 109B, 0000,003C,F9F0,2C00,0000,11F7  ;2953  q_id[3e]  ; Get the number of MS780-Es found
U 11F7, 0819,2014,0580,0800,0000,11F8  ;2954  d_q+k[.1]  ; Increment this counter
U 11F8, 0000,003C,F980,3C00,0000,11F9  ;2955  id[3e]_d  ; Save the counter
U 11F9, 0018,0038,11C0,0800,0000,11FA  ;2956  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2957      ; ...were the I/O base addresses will be stored
U 11FA, 081D,2000,7980,0800,0000,11FB  ;2958  d_q-d,k[.30]  ; ... adjust pointer
U 11FB, 0819,0014,7980,0800,0000,11FC  ;2959  d_d+k[.30]  ; Point the SC to the ID register
U 11FC, 0000,003C,F580,0800,0000,11FD  ;2960  k[.a]   ; ...to receive I/O base address
U 11FD, 0019,0014,F580,0800,0082,11FE  ;2961  alu_d+k[.a],sc_alu ; ...
U 11FE, 0810,0038,0180,0970,0000,11FF  ;2962  d_rc[0e]  ; Get base address to d
U 11FF, 0000,003C,0180,3400,0000,1200  ;2963  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2964 ;
        ;2965 ; Try next tr
        ;2966 ;
        ;2967 FIND.MEM.LP1:
U 1200, 0818,0014,0580,0A80,0000,1201  ;2968  r[0]_ia+k[.1],d_alu ; Increment TR level
        ;2969  alu_d.and.k[.f],
U 1201, 0019,0034,6180,0800,0010,1202  ;2970      clk.ubcc  ; Check if all done
U 1202, 0000,013C,0180,0800,0000,109C  ;2971  z?   ; Branch if yes
U 109C, 0000,003C,0180,0800,0000,108C  ;2972 =0 j/find.mem.loop  ; Try next TR
U 109D, 0810,0038,0180,0908,0010,1203  ;2973  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1203, 0000,013C,0180,0800,0000,109E  ;2974  z?   ; Check condition codes
U 109E, 0001,003E,0180,09F0,0000,0002  ;2975 =0 return2,rc[0e]_d ; Yes MS780E was found
U 109F, 0000,003E,0180,0800,0000,0001  ;2976  return1   ; No MS780E found
```

```
            ;2977  .PAGE
            ;2978  .TOC "   Generate IO Address"
            ;2979 ;++
            ;2980 ; FUNCTIONAL DESCRIPTION:
            ;2981 ;
            ;2982 ; This routine is used to generate an SBI Configuration Register
            ;2983 ; address from a TR level.
            ;2984 ;
            ;2985 ; CALLING CONSTRAINT:
            ;2986 ;
            ;2987 ; =0
            ;2988 ;
            ;2989 ; INPUTS:
            ;2990 ;
            ;2991 ; D - Contains TR level
            ;2992 ;
            ;2993 ; OUTPUTS:
            ;2994 ;
            ;2995 ; D - Contains SBI IO address
            ;2996 ;--
            ;2997 GENERATE.IO:
U 1204, 0000,003C,89F8,0800,0084,7205  ;2998  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1205, 0D00,003C,8D80,0800,0084,7206  ;2999  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 1206, 0000,003C,0980,0800,0084,B207  ;3000  sc_sc-k[.2]  ; insert bit 29
U 1207, 0801,001E,0180,0800,0000,0001  ;3001  d_d.ornot.mask,return1 ; and return
```

```
            ;3002  .PAGE
            ;3003  .TOC "    Set Starting Address"
            ;3004 ;++
            ;3005 ; FUNCTIONAL DESCRIPTION:
            ;3006 ;
            ;3007 ; This routine is used to set the starting address of the
            ;3008 ; memory to the specified value.
            ;3009 ;
            ;3010 ; CALLING CONSTRAINT:
            ;3011 ;
            ;3012 ; =0
            ;3013 ;
            ;3014 ; INPUTS:
            ;3015 ;
            ;3016 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3017 ;     (B Register Image divided by 2**16)
            ;3018 ; rc[0e] - Contains Address of Register A
            ;3019 ;
            ;3020 ; OUTPUTS:
            ;3021 ;
            ;3022 ; d - Contains aligned starting address (B Register Image)
            ;3023 ;
            ;3024 ; SIDE EFFECTS:
            ;3025 ;
            ;3026 ; d,q,va, and sc are destroyed
            ;3027 ;--
            ;3028 SET.START.ADDR:
U 1208, 0010,0038,6580,0970,0284,7209  ;3029  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1209, 0000,003C,01F8,0803,0000,120A  ;3030  va_va+4,q_0  ; Set address to Register B
            ;3031  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 120A, 0D00,003C,0980,0800,0084,B20B  ;3032       sc_sc-k[.2]  ; Setup to insert bit 14
U 120B, 0801,001C,0180,0800,0000,120C  ;3033  d_d.ornot.mask  ; Insert Write Enable bit
U 120C, 0000,003E,0180,A800,0000,0001  ;3034  cache.p_d[long],return1 ; Set the starting address and return
```

```
;3035  .PAGE
;3036  .TOC "   ENABLE NEXT MS780-E"
;3037
```

```
            ;3038  .PAGE
            ;3039  .TOC "   ENABLE NEXT MS780-E"
            ;3040 ;++
            ;3041 ; FUNCTIONAL DESCRIPTION:
            ;3042 ;
            ;3043 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;3044 ;    They will be tested  separately, according to the TR level at which
            ;3045 ;    they are located,  starting with the one  at the lowest level first.
            ;3046 ;          When  a test has finished  with the first unit, it will have to call
            ;3047 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3048 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3049 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3050 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3051 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3052 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3053 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3054 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3055 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3056 ;
            ;3057 ; CALLING CONSTRAINT:
            ;3058 ;
            ;3059 ;  =00
            ;3060 ;
            ;3061 ;--
            ;3062 ENABLE.NEXT.MS780E:
U 120D, 0000,003C,F9F0,2C00,0000,120E  ;3063  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3064  ALU_Q,   ; Check to see if it is zero
U 120E, 0001,203C,0180,0800,0010,120F  ;3065     CLK.UBCC  ; Check Condition Codes
U 120F, 0000,013C,0180,0800,0000,10A0  ;3066  Z?   ; ...for zero
U 10A0, 0000,003C,0180,0800,0000,1210  ;3067 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10A1, 0000,003E,0180,0800,0000,0002  ;3068  RETURN2   ; Zero No more units to test
            ;3069 ENABLE.NEXT:
U 1210, 0818,0038,C180,0800,0000,10A2  ;3070  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10A2, 0000,003D,0180,0800,0000,1208  ;3071 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10A3, 0818,0038,7980,0800,0000,1211  ;3072  D_K[.30]  ; Prepare to point to the ID register holding
            ;3073     ; ...the I/O Base address of the next MS780-E
U 1211, 0819,0014,1180,0800,0000,1212  ;3074  D_D+K[.4]  ; ...
U 1212, 0000,003C,F580,0800,0000,1213  ;3075  K[.A]   ; ...
U 1213, 0819,0014,F580,0800,0000,1214  ;3076  D_D+K[.A]  ; ...
U 1214, 0000,003C,F9F0,2C00,0000,1215  ;3077  Q_ID[3E]  ; Get MS780-E Counter
            ;3078  D_D-Q,   ; D now holds the pointer to the ID register
U 1215, 081D,0000,0180,0800,0082,1216  ;3079    SC_ALU  ; ...
U 1216, 0000,003C,01F0,2400,0000,1217  ;3080  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1217, 0001,203C,0180,09F0,0000,1218  ;3081  RC[0E]_Q  ; Save it also in RC[0E]
U 1218, 0F03,003C,0180,09E8,0000,10A4  ;3082  RC[0D]_0,D_0  ; Set starting address to zero
U 10A4, 0000,003D,0180,0800,0000,1208  ;3083 =0 CALL[SET.START.ADDR] ; ....
U 10A5, 0F00,003C,F9F0,2C00,0000,1219  ;3084  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1219, 081D,2008,0180,0800,0000,121A  ;3085  D_Q-D-1   ; Subtract 1 from total
U 121A, 0000,003C,F980,3C00,0000,10A6  ;3086  ID[3E]_D  ; Adjust the pointer
U 10A6, 0000,003D,0180,0800,0000,11B7  ;3087 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10A7, 0000,003E,0180,0800,0000,0001  ;3088  RETURN1   ; Tell caller another unit was found
            ;3089
```

```
            ;3090  .PAGE
            ;3091  .TOC "    Set ECC Bits"
            ;3092 ;++
            ;3093 ; FUNCTIONAL DESCRIPTION:
            ;3094 ;
            ;3095 ; This routine is used to set the specified ECC bits
            ;3096 ; in Register B. Other bits in the register remain unchanged.
            ;3097 ;
            ;3098 ; CALLING CONSTRAINT:
            ;3099 ;
            ;3100 ; =0
            ;3101 ;
            ;3102 ; INPUTS:
            ;3103 ;
            ;3104 ; d - Contains the bits to set in <6:0>
            ;3105 ; rc[0] - Contains base address of controller
            ;3106 ;
            ;3107 ; SIDE EFFECTS:
            ;3108 ;
            ;3109 ; d,va,sc are destroyed
            ;3110 ;--
            ;3111 SET_ECC:
U 121B, 0010,0038,0180,0970,0200,121C  ;3112  va_rc[0E]  ; Set address of Register B
U 121C, 0000,003C,0180,0803,0000,121D  ;3113  va_va+4    ; ...in VA
U 121D, 0000,003C,01E0,C800,0000,121E  ;3114  q_d,d[long]_cache.p ; Read current contents of register
U 121E, 0819,0024,5980,0800,0000,121F  ;3115  d_d.andnot.k[.7f] ; Clear current ECC bits
U 121F, 081D,0030,0180,0800,0000,1220  ;3116  d_d.or.q  ; Insert new ecc bits
U 1220, 0000,003E,0180,A800,0000,0001  ;3117  cache.p_d[long],return1 ; Set the specified bits and return
            ;3118
            ;3119
```

```
            ;3120  .PAGE
            ;3121  .TOC "   MS780-E/H CNFG REG CLEANUP"
            ;3122 ;*************************************************************
            ;3123 ;++
            ;3124 ;
            ;3125 ; Functional Description:
            ;3126 ; ----------------------
            ;3127 ;
            ;3128 ;      This subroutine is used to clear all error conditions
            ;3129 ; in the MS780-E/H Configuration/Status/Error Registers.
            ;3130 ; Bits beeing cleared are:
            ;3131 ;
            ;3132 ; CNFG Regsiter B <11:00>
            ;3133 ;
            ;3134 ; CNFG Register C and D <31:28> and <10:06>
            ;3135 ;   ---
            ;3136 ;
            ;3137 ;
            ;3138 ; Calling Constraint: =0
            ;3139 ; ------------------
            ;3140 ;
            ;3141 ;
            ;3142 ; Inputs:
            ;3143 ; ------
            ;3144 ;
            ;3145 ; RC[0E] MUST contain the I/O base address of
            ;3146 ; the MS780-E/H currently under test
            ;3147 ;
            ;3148 ; Outputs:
            ;3149 ; -------
            ;3150 ;
            ;3151 ; NO specific outputs.
            ;3152 ; Above mentioned bits will be cleared.
            ;3153 ;
            ;3154 ; Side Effects:
            ;3155 ; ------------
            ;3156 ;
            ;3157 ; The contents of the following registers will be lost
            ;3158 ; Q, D, SC, VA
            ;3159 ;
            ;3160 ;
            ;3161 ;--
            ;3162 ;*************************************************************
            ;3163 MS780E.CLEANUP:
U 1221, 0010,0038,0180,0970,0200,1222  ;3164  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
            ;3165  D_0,     ; Get data to clear Read/Write bits
U 1222, 0F00,003C,0180,0803,0000,1223  ;3166   VA_VA+4    ; Point to CNFG Reg B
U 1223, 0000,003C,0180,A800,0000,1224  ;3167  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 1224, 0818,0038,7980,0800,0000,1225  ;3168  D_K[.30]    ; Prepare to form x30000780 to clear
            ;3169       ; ...CNFG Reg's C and D
U 1225, 0B00,003C,0180,0800,0000,1226  ;3170  D_D.SWAP    ; D = x30000000
            ;3171  Q_D,     ; Save in the Q Reg
U 1226, 0818,0038,CDE0,0800,0000,1227  ;3172   D_K[.F0]   ; D = xF0
U 1227, 0000,003C,0D80,0800,0084,7228  ;3173  SC_K[.3]    ; Must shift D Reg left 3 bits to get
            ;3174       ; ...x780
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1797
ESKAR52.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:56                   Page   9

U 1228. 0D00.003C.0180.0800.0000.1229  ;3175  D_DAL.SC    ; D = x780
        ;3176  D_D.OR.Q,    ; D = x30000780
U 1229. 081D.0030.0180.0803.0000.122A  ;3177   VA_VA+4    ; Point to CNFG Reg C
U 122A. 0000.003C.0180.A800.0000.122B  ;3178  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 122B. 0000.003C.0180.0800.0000.122C  ;3179  NOP
U 122C. 0000.003C.0180.A800.0000.122D  ;3180  CACHE.P_D[LONG]   ; Do second write to make sure that
        ;3181       ; ...uSequencer Parity error bit
        ;3182       ; ...is clear
U 122D. 0000.003C.0180.0803.0000.122E  ;3183  VA_VA+4    ; Point to CNFG Reg D
U 122E. 0000.003C.0180.A800.0000.122F  ;3184  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 122F. 0000.003C.0180.0800.0000.1230  ;3185  NOP
        ;3186  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
        ;3187       ; ...Parity error bit
U 1230. 0000.003E.0180.A800.0000.0001  ;3188   RETURN1    ; Return to caller
        ;3189
```

```
            ;3190  .PAGE
            ;3191  .TOC "   CHECK SBI.ERR"
            ;3192 ;++
            ;3193 ; FUNCTIONAL DESCRIPTION:
            ;3194 ;
            ;3195 ; This subroutine can be used to see whether the SBI.ERR Register
            ;3196 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;3197 ; routine will have to pass to this subroutine in the SC register
            ;3198 ; the number of the error bit to checked.
            ;3199 ;
            ;3200 ; CALLING CONSTRAINT:
            ;3201 ;
            ;3202 ; =00
            ;3203 ;
            ;3204 ; INPUTS:
            ;3205 ;
            ;3206 ; SC with the number of the error bit to be checked
            ;3207 ;
            ;3208 ; OUTPUTS:
            ;3209 ;
            ;3210 ; RC[08]-Contains the SBI.ERROR Register
            ;3211 ;--
            ;3212 CHECK_SBI_ERR:
U 1231, 0000,003C,65F0,2C00,0000,1232  ;3213  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 1232, 0001,203C,0180,09C0,0000,1233  ;3214  RC[08]_Q  ; Save for error logout
            ;3215  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 1233, 0001,2024,0180,0800,0010,1234  ;3216   CLK.UBCC ; ...
U 1234, 0000,013C,0180,0800,0000,10A8  ;3217  Z?   ; Check condition codes
U 10A8, 0000,003E,0180,0800,0000,0002  ;3218 =0 RETURN2   ; Bit is set
U 10A9, 0000,003E,0180,0800,0000,0001  ;3219  RETURN1   ; Bit is not set
            ;3220
```

```
          ;3221  .PAGE
          ;3222  .TOC "    64K OR 256K CHIPS"
          ;3223 ;*****************************************************
          ;3224 ;++
          ;3225 ;
          ;3226 ; Functional Description:
          ;3227 ; ----------------------
          ;3228 ; This subroutine is used to find the type of chips
          ;3229 ; on the MS780-E/H arrays.  The RETURN modes are as
          ;3230 ; follows:
          ;3231 ;
          ;3232 ; RETURN1 = Error. Mixed 64K and 256K arrays
          ;3233 ;
          ;3234 ; RETURN2 = 64K chips on the array
          ;3235 ;
          ;3236 ; RETURN3 = 256K chips on the array
          ;3237 ;
          ;3238 ; Calling Constraint: =00
          ;3239 ; ------------------
          ;3240 ;
          ;3241 ;
          ;3242 ; Inputs:
          ;3243 ; ------
          ;3244 ;
          ;3245 ; RC[0E] must hold the I/O base address of the MS780-E/H
          ;3246 ; currently under test
          ;3247 ;
          ;3248 ; Outputs:
          ;3249 ; -------
          ;3250 ;
          ;3251 ; NO specific outputs. (See RETURN modes above)
          ;3252 ;
          ;3253 ;
          ;3254 ; Side Effects:
          ;3255 ; ------------
          ;3256 ;
          ;3257 ; The contents of the following registers will be lost:
          ;3258 ;
          ;3259 ; D
          ;3260 ;
          ;3261 ;--
          ;3262 ;*****************************************************
          ;3263 64K.OR.256K:
U 1235.  0010,0038,0180,0970,0200,1236  ;3264  VA_RC[0E]    ; Point to CNFG Register A
U 1236.  0000,003C,0180,C800,0000,1237  ;3265  D[LONG]_CACHE.P    ; Read the register
U 1237.  0200,003C,0180,0800,0000,1238  ;3266  D_D.RIGHT2    ; Shift received contents two bits
          ;3267          ; ...to the right
U 1238.  0600,003C,0180,0800,0000,1239  ;3268  D_D.RIGHT    ; Shift D one more time
U 1239.  0000,193C,0180,0800,0000,102C  ;3269  D1-0?    ; Branch on the RAM Type
U 102C.  0000,003E,0180,0800,0000,0001  ;3270 =1100 RETURN1    ; Error: both 64K and 256K RAMs present
U 102D.  0000,003E,0180,0800,0000,0002  ;3271  RETURN2    ; 64K RAMs found
U 102E.  0000,003E,0180,0800,0000,0003  ;3272  RETURN3    ; 256K RAMs found
U 102F.  0000,003E,0180,0800,0000,0001  ;3273  RETURN1    ; Error: missing arrays
          ;3274
```

```
            ;3275  .PAGE
            ;3276  .TOC "    GET MEMORY SIZE"
            ;3277 ;*************************************************************
            ;3278 ;**
            ;3279 ;
            ;3280 ; Functional Description:
            ;3281 ; ----------------------
            ;3282 ;
            ;3283 ; This subroutine can be used to get the size of the memory.
            ;3284 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3285 ; increments.
            ;3286 ;
            ;3287 ; Calling Constraint: =0
            ;3288 ; ------------------
            ;3289 ;
            ;3290 ;
            ;3291 ; Inputs:
            ;3292 ; ------
            ;3293 ;
            ;3294 ; RC[0E] Must have the Address of CNFG Register A
            ;3295 ;
            ;3296 ; Outputs:
            ;3297 ; -------
            ;3298 ;
            ;3299 ; Register D will contain upon exit the size of the
            ;3300 ; memory aligned on Mega-byte boundary.
            ;3301 ;
            ;3302 ; Side Effects:
            ;3303 ; ------------
            ;3304 ;
            ;3305 ; The contents of the following registers will be lost:
            ;3306 ;
            ;3307 ; D, SC, Q
            ;3308 ;
            ;3309 ;--
            ;3310 ;*************************************************************
            ;3311 GET.MEMORY.SIZE:
U 123A, 0010,0038,0180,0970,0200,123B  ;3312  VA_RC[0E]    ; Point to CNFG Reg A
U 123B, 0000,003C,0180,C800,0000,123C  ;3313  D[LONG]_CACHE.P    ; Read the register
U 123C, 001B,0000,D980,0800,0082,123D  ;3314  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 123D, 0D00,003C,0180,0800,0000,123E  ;3315  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 123E, 0819,0034,5580,0800,0000,1030  ;3316  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3317 =00 CALL[64K.OR.256K],  ; Find out the size of the arrays
U 1030, 0000,003D,01E0,0800,0000,1235  ;3318    Q_D    ; Save Memory size bits in Q
U 1031, 0000,003D,0D80,0800,0084,7058  ;3319  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3320       ; ...arrays on the controller
            ;3321 ;
            ;3322 ; The duplication of the next two return status is necessary so the returning
            ;3323 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3324 ; modifications.
            ;3325 ;
U 1032, 0819,2014,0580,0800,0000,1033  ;3326  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3327       ; ...(found 1 Meg arrays)
U 1033, 0819,2014,0580,0800,0000,123F  ;3328  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3329       ; ...(found 4 Meg arrays)
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1801
ESKAR52.MCR                        MICRO2  1P(00)    26-JUL-85  17:05:56                 Page   9

U 123F. 0000,003C,21F8,0800,0084,7240  ;3330  SC_K[.14],Q_0    ; Prepare to shift to Megabyte position
          ;3331  D_DAL.SC,    ; Shift bits <5:0> to <25:20>
U 1240. 0D00,003E,0180,0800,0000,0001  ;3332   RETURN1    ; Return with memory size in Register D
          ;3333
```

```
;3334  .PAGE
;3335  .TOC "   TYPE ASCII MESSAGE ROUTINE"
;3336 ;*************************************************************
;3337 ;++
;3338 ;
;3339 ; Functional Description:
;3340 ; ----------------------
;3341 ;
;3342 ; This routine is used to type ASCII messages on
;3343 ; the console terminal.
;3344 ;
;3345 ;
;3346 ; Calling Constraint: =0
;3347 ; ------------------
;3348 ;
;3349 ;
;3350 ; Inputs:
;3351 ; ------
;3352 ;
;3353 ; The D Register must hold the number of the message
;3354 ; desired to be typed.
;3355 ;
;3356 ; Outputs:
;3357 ; -------
;3358 ;
;3359 ; NONE
;3360 ;
;3361 ;
;3362 ; Side Effects:
;3363 ; ------------
;3364 ;
;3365 ; The contents of the following registers will be lost:
;3366 ;
;3367 ; D, SC
;3368 ;
;3369 ;--
;3370 ;*************************************************************
;3371 MSGCOM:
;3372  SC_K[.A],   ; Get function code for Mic Mon
U 1241, 0000,003C,F580,0800,0084,7249  ;3373   J/COMMON  ; Go to the common portion of code
;3374
```

```
;3375  .PAGE
;3376  .TOC "   TYPE ASCII MESSAGE WITH CRLF"
;3377 ;*************************************************************
;3378 ;++
;3379 ;
;3380 ; Functional Description:
;3381 ; ----------------------
;3382 ;
;3383 ; This routine is used to type ASCII messages on the
;3384 ; console terminal.  The message is follwed by a Cerriage
;3385 ; Return and a Line Feed.
;3386 ;
;3387 ;
;3388 ; Calling Constraint: =0
;3389 ; ------------------
;3390 ;
;3391 ;
;3392 ; Inputs:
;3393 ; ------
;3394 ;
;3395 ; The D register must hold the number of the message
;3396 ; to be printed.
;3397 ;
;3398 ; Outputs:
;3399 ; -------
;3400 ;
;3401 ; NONE
;3402 ;
;3403 ;
;3404 ; Side Effects:
;3405 ; ------------
;3406 ;
;3407 ; The contents of the follwoing registers will be lost:
;3408 ;
;3409 ; D, Q, SC
;3410 ;
;3411 ;--
;3412 ;*************************************************************
;3413 MSGCOM_CRLF:
;3414   Q_K[.9].LEFT,SI/ZERO,  ; Get Function code for Mic Mon
;3415    SC_ALU,    ; ...
U 1242, 0038,0038,D9C0,0800,0082,1249  ;3416   J/COMMON  ; ...Go to the common code
;3417
```

```
            ;3418  .PAGE
            ;3419  .TOC "    TYPE NUMERIC ROUTINE"
            ;3420 ;*************************************************************
            ;3421 ;++
            ;3422 ;
            ;3423 ; Functional Description:
            ;3424 ; ----------------------
            ;3425 ;
            ;3426 ; This subroutine is used for typing numerical data
            ;3427 ; on the console terminal.
            ;3428 ;
            ;3429 ; Calling Constraint: =0
            ;3430 ; ------------------
            ;3431 ;
            ;3432 ;
            ;3433 ; Inputs:
            ;3434 ; ------
            ;3435 ;
            ;3436 ; Register D must hold the numeric data to be typed.
            ;3437 ;
            ;3438 ;
            ;3439 ; Outputs:
            ;3440 ; -------
            ;3441 ;
            ;3442 ; NONE
            ;3443 ;
            ;3444 ;
            ;3445 ; Side Effects:
            ;3446 ; ------------
            ;3447 ;
            ;3448 ; The contents of the following registers will be lost:
            ;3449 ;
            ;3450 ; D, SC, ID[29]
            ;3451 ;
            ;3452 ;--
            ;3453 ;*************************************************************
            ;3454 RCECOM:
            ;3455  SC_K[.10],   ; Get Function Code for Mic Mon
U 1243, 0000,003C,6580,0800,0084,7245  ;3456   J/RCE.COMMON  ; ...Go to the common code
            ;3457
```

```
;3458  .PAGE
;3459  .TOC "    TYPE NUMERIC WITH CRLF"
;3460 ;*************************************************************
;3461 ;++
;3462 ;
;3463 ; Functional Description:
;3464 ; ----------------------
;3465 ;
;3466 ; This routine is used to type numeric data on the
;3467 ; console terminal.  The data is followed by a Carriage
;3468 ; Return and a Line Feed.
;3469 ;
;3470 ;
;3471 ; Calling Constraint: -0
;3472 ; ------------------
;3473 ;
;3474 ;
;3475 ; Inputs:
;3476 ; ------
;3477 ;
;3478 ; The D register must hold the numeric data to be typed.
;3479 ;
;3480 ;
;3481 ; Outputs:
;3482 ; -------
;3483 ;
;3484 ; NONE
;3485 ;
;3486 ;
;3487 ; Side Effects:
;3488 ; ------------
;3489 ;
;3490 ; The contents of the following registers will be lost:
;3491 ;
;3492 ; D, SC, ID[29]
;3493 ;
;3494 ;--
;3495 ;*************************************************************
;3496 RCECOM_CRLF:
;3497  SC_K[.14],    ; Get function code for Mic Mon
U 1244, 0000,003C,2180,0800,0084,7245  ;3498   J/RCE.COMMON  ; ...Go to the common code
;3499
```

```
            ;3500  .PAGE
            ;3501  .TOC "   COMMON CODE FOR TYPE NUMERIC ROUTINES"
            ;3502 ;*****************************************************
            ;3503 ;++
            ;3504 ;
            ;3505 ; Functional Description:
            ;3506 ; ----------------------
            ;3507 ;
            ;3508 ; This portion of code will be used by the routines that type
            ;3509 ; numeric data to the console terminal.
            ;3510 ; Its function is to save the contents of register RC[0E]
            ;3511 ; and replace them with the data to be typed.
            ;3512 ; The old contents of register RC[0E] will be reinstated before
            ;3513 ; the control is returned to the caller.
            ;3514 ;
            ;3515 ; Calling Constraint: =0
            ;3516 ; ------------------
            ;3517 ;
            ;3518 ;
            ;3519 ; Inputs:
            ;3520 ; ------
            ;3521 ;
            ;3522 ; As setup by RCECOM and RCECOM_CRLF ROUTINES.
            ;3523 ;
            ;3524 ;
            ;3525 ; Outputs:
            ;3526 ; -------
            ;3527 ;
            ;3528 ; NONE
            ;3529 ;
            ;3530 ;
            ;3531 ; Side Effects:
            ;3532 ; ------------
            ;3533 ;
            ;3534 ; Contents of teh following registers will be lost:
            ;3535 ;
            ;3536 ; D, Q, SC, ID[29]
            ;3537 ;
            ;3538 ;--
            ;3539 ;*****************************************************
            ;3540 RCE.COMMON:
U 1245, 0000,003C,01E0,0800,0000,1246  ;3541  Q_D     ; Save data to be typed in Q Register
U 1246, 0810,0038,0180,0970,0000,1247  ;3542  D_RC[0E]    ; Get contents of RC[0E]
            ;3543  ID[29]_D,    ; Save contents of RC[0E] in ID
            ;3544        ; ...register 29
U 1247, 0001,203C,A580,3DF0,0000,10AA  ;3545   RC[0E]_Q  ; Get data to be typed in RC[0E]
U 10AA, 0000,003D,0180,0800,0000,1249  ;3546 =0 CALL[COMMON]    ; CALL the common code
U 10AB, 0000,003C,A5F0,2C00,0000,1248  ;3547  Q_ID[29]    ; Retrieve saved contents
            ;3548       ; ...of RC[0E]
            ;3549  RC[0E]_Q,    ; Reinstate RC[0E]
U 1248, 1001,203E,0180,09F0,0000,0001  ;3550   RETURN1    ; ...and return to caller
            ;3551
```

```
            ;3552  .PAGE
            ;3553  .TOC "   COMMON CODE FOR TYPE ASCII AND NUMERIC"
            ;3554 ;*************************************************************
            ;3555 ;++
            ;3556 ;
            ;3557 ; Functional Description:
            ;3558 ; ----------------------
            ;3559 ;
            ;3560 ; This code is used by both Type-ASCII and Type-Numeric
            ;3561 ; routines.  It provides the interface to the Micro Monitor.
            ;3562 ;
            ;3563 ;
            ;3564 ; Calling Constraint: =0
            ;3565 ; ------------------
            ;3566 ;
            ;3567 ;
            ;3568 ; Inputs:
            ;3569 ; ------
            ;3570 ;
            ;3571 ; As set up by MSGCOM, MSGCOM_CRLF, RCECOM and RCECOM_CRLF.
            ;3572 ;
            ;3573 ;
            ;3574 ; Outputs:
            ;3575 ; -------
            ;3576 ;
            ;3577 ; NONE
            ;3578 ;
            ;3579 ;
            ;3580 ; Side Effects:
            ;3581 ; ------------
            ;3582 ;
            ;3583 ; The contents of the following registers will be lost:
            ;3584 ;
            ;3585 ; D, Q, SC, ID[29]
            ;3586 ;
            ;3587 ;--
            ;3588 ;*************************************************************
            ;3589 COMMON:
U 1249, 0000,003C,0180,0800,0100,124A  ;3590  FE_SC    ; Save the Function Code for Mic Mon
            ;3591  SC_K[.9],    ; Get ready to shift message number
U 124A, 0000,003C,D9F8,0800,0084,724B  ;3592   Q_0    ; ...(or type ASCII)
U 124B, 0D00,003C,0180,0800,0000,124C  ;3593  D_DAL.SC   ; Shift
            ;3594 CNFCOM:
U 124C, 0000,003C,C1F0,2C00,0000,124D  ;3595  Q_ID[T0]   ; Get the Type-Out flag
            ;3596  ALU_Q,CLK.UBCC,    ; See if the Type-Out flag is set.
U 124D, 0C01,203C,01E0,0800,0010,124E  ;3597   Q_D,D_Q   ; Get Type-Out flag to D
            ;3598  ALU.N?,     ; Is the Type-Out flag set?
U 124E, 0819,1B14,0580,0800,0000,1037  ;3599   D_D+K[.1]  ; Clear Type-Out flag
            ;3600 =0111 RETURN1,    ; Type-Out flag was NOT set
U 1037, 0C00,003E,0180,0800,0000,0001  ;3601   D_Q   ;
            ;3602  ID[T0]_D,    ; Save Type-Out flag
            ;3603   D_Q,    ; Get number of the messge to
            ;3604       ; ...print
U 103F, 0C00,003C,C180,3C00,0081,124F  ;3605   SC_FE   ; Get the Function Code for Mic Mon
            ;3606  D_D.OR.K[SC],    ; Get Function Code in the D Reg
```

```
U 124F, 0819,0030,1D80,0800,0000,105E  ;3607   J/CALLCON  ; Call Mic Mon
          ;3608
```

```
            ;3609  .PAGE
            ;3610  .TOC "   DECREMENT TO ROUTINE"
            ;3611 ;*****************************************************************
            ;3612 ;++
            ;3613 ;
            ;3614 ; Functional Description:
            ;3615 ; ----------------------
            ;3616 ;
            ;3617 ; This routine is used to set the Type-Out flag.
            ;3618 ;
            ;3619 ; Calling Constraint:
            ;3620 ; ------------------
            ;3621 ;
            ;3622 ;
            ;3623 ; Inputs:
            ;3624 ; ------
            ;3625 ;
            ;3626 ; SC must hold the number of messages to be
            ;3627 ;
            ;3628 ;
            ;3629 ; Outputs:
            ;3630 ; -------
            ;3631 ;
            ;3632 ; T0 is loaded with 0 - (SC)
            ;3633 ;
            ;3634 ;
            ;3635 ; Side Effects:
            ;3636 ; ------------
            ;3637 ;
            ;3638 ; The contents of the following regsiters will be lost
            ;3639 ;
            ;3640 ; D, SC, Q, T0
            ;3641 ;
            ;3642 ;--
            ;3643 ;*****************************************************************
            ;3644 DECREMENT_T0:
U 1250, 0F00,003C,0180,0800,0000,1251  ;3645  D_0     ; Clear the D Register
U 1251, 0819,0000,1D80,0800,0000,1252  ;3646  D_D-K[SC]    ; Decrement by the specified amount
            ;3647  ID[T0]_D,    ; Save in T0
U 1252, 0000,003E,C180,3C00,0000,0001  ;3648   RETURN1    ; Return to caller
            ;3649
```

```
;3650  .PAGE
;3651  .TOC "    ASCII MESSAGES"
;3652
;3653 ;& 01,@MS780-E/H IO BASE ADDRESS = @
;3654 ;& 02,/LOWER CONTROLLER /
;3655 ;& 03,/UPPER CONTROLLER /
;3656 ;& 04,/MAX ADDR + 1 = /
;3657 ;& 05,/  BOARD NUMBER = /
;3658 ;& 06,/     NUMBER OF CRD ERRORS = /
;3659
;3660
```

```
;3661   .PAGE "TEST 1F7 SINGLE BIT ERROR ON MASKED WRITES"
;3662  ;*************************************************************
;3663  ;+++
;3664  ;
;3665  ; Functional Description:
;3666  ; ----------------------
;3667  ;
;3668  ; This test forces CRD errors on Masked Writes, and verifies whether the
;3669  ; error is detected and properly corrected.  The following cases will be
;3670  ; tested:
;3671  ;
;3672  ;     i.  Error in byte 3
;3673  ;    ii.  Error in byte 2
;3674  ;   iii.  Error in byte 1
;3675  ;    iv.  Error in byte 0
;3676  ;     v.  Error in check bits
;3677  ;    vi.  Error in byte 0
;3678  ;
;3679  ; Cases i through iv force errors in bytes that are not being written
;3680  ; with new data and a check is made to see that the data was corrected.
;3681  ; Case v forces an error in the check bits.  It is expected that the data
;3682  ; will not be modified, while the check bits will be corrected when the
;3683  ; data is written to the array.  Finally, case vi forces a CRD error in
;3684  ; one byte that should be written over with new data.  A read of the
;3685  ; array will assure that the byte was indeed written over.  The above
;3686  ; cases will make up the subtests below.
;3687  ;
;3688  ;   [Subtest 1:]
;3689  ;
;3690  ; Location 0 is initialize to x01000000, the substitute ECC bits are set
;3691  ; to xC (for data of all zeroes) in Diagnostic Mode 2 a write is done to
;3692  ; bytes 0, 1, and 2, thus forcing the CRD error.  On a subsequent
;3693  ; extended read of the location in Diagnostic Mode 3 it is expected that
;3694  ; the data returned from the array will be all zeroes.
;3695  ;
;3696  ;   [Subtest 2:]
;3697  ;
;3698  ; In this case location 0 is initialized to x10000 and a write is done to
;3699  ; bytes 0 and 1 (everything else will be the same as in the previous
;3700  ; subtest, ie., setting up the diagnostic modes and the data returned
;3701  ; from memory).
;3702  ;
;3703  ;   [Subtest 3:]
;3704  ;
;3705  ; This subtest initializes location 0 to x100 and a write is done to byte
;3706  ; 0.  Diagnostic modes and expected data returned from the memory are the
;3707  ; save as in previous subtests.
;3708  ;
;3709  ;   [Subtest 4:]
;3710  ;
;3711  ; Location 0 is initialized to x01 and a write is done to bytes 1, 2, and
;3712  ; 3.
;3713  ;
;3714  ;   [Subtest 5:]
;3715  ;
```

```
;3716 ; This subtest initializes the array location 0 too 0, sets the substi-
;3717 ; tute ECC bits to x0D and in Diagnostic Mode 2 executes a masked write
;3718 ; to byte 0.  It is expected that the data on the array was not modified
;3719 ; (ie., it is still 0) and that the check bits on the array will be for
;3720 ; data of all 0's.
;3721 ;
;3722 ;   [Subtest 6:]
;3723 ;
;3724 ; In this subtest location 0 will be initialized to x03.  The substitute
;3725 ; ECC bits are set for data of one (x43), and in Diagnostic mode 2 byte 0
;3726 ; will be written with data of 0.  this will force a CRD error in byte 0
;3727 ; but the new data should still be written to byte 0.
;3728 ;
;3729 ; Logic Description:
;3730 ; -----------------
;3731 ;  N/A
;3732 ;
;3733 ; Error Logout:
;3734 ; ------------
;3735 ;
;3736 ; See individual error reports.
;3737 ;
;3738 ; Sync Point Description:
;3739 ; ----------------------
;3740 ;  N/A
;3741 ;
;3742 ;;;
```

```
U 10AD, 0000,003C,0180,0800,0000,1038  ;3771  NOP
U 1038, 0000,003D,0180,0800,0000,108A  ;3772 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1039, 0000,003C,0180,0800,0000,128F  ;3773  J/T.44.EXIT  ; No MS780-E's to test, exit this test
U 103A, 0000,003C,0180,0800,0000,1040  ;3774  NOP
          ;3775 =
          ;3776  ;
          ;3777  ; This is the entry point for restarting the test after one controller
          ;3778  ; has been tested and the next controller is to be configured and
          ;3779  ; tested.  If there is no more controllers then exit this test.
          ;3780  ;
          ;3781 =00
          ;3782 RESTART.TEST.44:
U 1040, 0000,003D,0180,0800,0000,106C  ;3783  CALL[START.TEST] ; Configure the controllers
U 1041, 0000,003C,0180,0800,0000,128F  ;3784  J/T.44.EXIT  ; No more controllers to configure, exit test
U 1042, 0000,003C,0180,0800,0000,10AE  ;3785  NOP
          ;3786 =
          ;3787 =0
          ;3788 T.44.S1:
U 10AE, 0000,003D,0180,0800,0000,119D  ;3789  SUBTEST   ; Update the subtest counter
U 10AF, 0000,003C,7D80,0800,0084,7253  ;3790  SC_K[.18]  ; Get ready to assert bit 24 in RC[0A]
U 1253, 0003,001C,0180,09D0,0000,1254  ;3791  RC[0A]_NOT.MASK  ; Load RC[0A] with x01000000 for forcing
          ;3792     ; an error to the test location
U 1254, 0003,003C,0180,09E0,0000,10B0  ;3793  RC[0C]_0  ; Load RC[0C] with the expected memory data
U 10B0, 0000,003D,0180,0800,0000,1195  ;3794 =0 ERLOOP   ; loop on error from here
U 10B1, 0000,003C,0180,0800,0000,10B2  ;3795  NOP
U 10B2, 0000,003D,0180,0800,0000,1084  ;3796 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10B3, 0018,0038,1980,0800,0200,1255  ;3797  VA_K[ZERO]  ; Point VA to test location 0
U 1255, 0810,0038,0180,0950,0000,1256  ;3798  D_RC[0A]  ; Load D with init pattern for write
U 1256, 0000,003C,0180,A800,0000,10B4  ;3799  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 10B4, 0818,0039,0980,0800,0000,11B0  ;3800 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 10B5, 0000,003C,0180,0800,0000,10B6  ;3801  NOP
U 10B6, 0818,0039,8580,0800,0000,121B  ;3802 =0 SET.ECC[.C]  ; Set substitute ECC mode
U 10B7, 0018,0038,0D80,0800,0200,1257  ;3803  VA_K[.3]  ; Load VA with x3 for writing byte 3
U 1257, 0F00,003C,0180,0800,0000,1258  ;3804  D_0   ; data to write in bytes 0,1,2
U 1258, 0000,003C,0180,3000,3000,10B8  ;3805  CACHE[SECOND.REF]_D ; Write the pattern while forcing the MSC
          ;3806     ; field to SECOND.REFERENCE
U 10B8, 0818,0039,0D80,0800,0000,11B0  ;3807 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 10B9, 0018,0038,1980,0800,0200,1259  ;3808  VA_K[ZERO]  ; Point VA to test location 0
U 1259, 0000,003C,0180,C800,0000,125A  ;3809  D[LONG]_CACHE.P  ; Read test location 0 and
U 125A, 0001,003C,0180,09D8,0000,125B  ;3810  RC[0B]_D  ; save in RC[0B]
U 125B, 0001,003C,0180,0800,0010,125C  ;3811  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 125C, 0000,013C,0180,0800,0000,10BA  ;3812  Z?   ; was the read data equal to 0?
U 10BA, 0000,003C,0180,0800,0000,10BC  ;3813 =0 J/T.44.S1.BAD.DATA ; No, Call an error
U 10BB, 0000,003C,0180,0800,0000,10BE  ;3814  J/T.44.S2  ; Yes, continue with the test
```

```
            ;3815  .PAGE
            ;3816 ;
            ;3817 ;////////////////////////////////////////////////////////////////////////////////
            ;3818 ;
            ;3819 ; Data read from test location was not equal to 0, byte 3 did not get
            ;3820 ; corrected.  Call an error.
            ;3821 ;
            ;3822 =0
            ;3823 T.44.S1.BAD.DATA:
U 10BC, 0000.003D.D580.0800.0084.705C  ;3824  ERROR2[.6]
U 10BD, 0000.003C.0180.0800.0000.10BE  ;3825  NOP
            ;3826 ;
            ;3827 ;////////////////////////////////////////////////////////////////////////////////
            ;3828 ;
            ;3829 ; ERROR LOGOUT:
            ;3830 ;
            ;3831 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3832 ;  I/O Address of Configuration Register C/D
            ;3833 ;  Expected Memory Data
            ;3834 ;  Received Memory Data
            ;3835 ;  Data Pattern to Force the Error
            ;3836 ;  Unused
            ;3837 ;
            ;3838 ;////////////////////////////////////////////////////////////////////////////////
            ;3839 ;
            ;3840 =0
            ;3841 T.44.S2:
U 10BE, 0000.003D.0180.0800.0000.119D  ;3842  SUBTEST   ; Update the subtest counter
U 10BF, 0000.003C.6580.0800.0084.725D  ;3843  SC_K[.10]  ; Get ready to assert bit 16 in RC[0A]
U 125D, 0003.001C.0180.09D0.0000.125E  ;3844  RC[0A]_NOT.MASK  ; Load RC[0A] with x00010000 for forcing
            ;3845      ; an error to the test location
U 125E, 0003.003C.0180.09E0.0000.10C0  ;3846  RC[0C]_0  ; Load RC[0C] with the expected memory data
U 10C0, 0000.003D.0180.0800.0000.1195  ;3847 =0 ERLOOP   ; loop on error from here
U 10C1, 0000.003C.0180.0800.0000.10C2  ;3848  NOP
U 10C2, 0000.003D.0180.0800.0000.1084  ;3849 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10C3, 0018.0038.1980.0800.0200.125F  ;3850  VA_K[ZERO]  ; Point VA to test location 0
U 125F, 0810.0038.0180.0950.0000.1260  ;3851  D_RC[0A]  ; Load D with init pattern for write
U 1260, 0000.003C.0180.A800.0000.10C4  ;3852  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 10C4, 0818.0039.0980.0800.0000.11B0  ;3853 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 10C5, 0000.003C.0180.0800.0000.10C6  ;3854  NOP
U 10C6, 0818.0039.8580.0800.0000.121B  ;3855 =0 SET.ECC[.C]  ; Set substitute ECC mode
U 10C7, 0018.0038.1980.0800.0200.1261  ;3856  VA_K[ZERO]  ; Load VA with 0 for writing bytes 1 and 2
U 1261, 0F00.003C.0180.0800.0000.1262  ;3857  D_0   ; data to write in bytes 0 and 1
U 1262, 0000.403C.0180.A800.0000.10C8  ;3858  CACHE.P_D[WORD]  ; Write the pattern with data type WORD
U 10C8, 0818.0039.0D80.0800.0000.11B0  ;3859 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 10C9, 0018.0038.1980.0800.0200.1263  ;3860  VA_K[ZERO]  ; Point VA to test location 0
U 1263, 0000.003C.0180.C800.0000.1264  ;3861  D[LONG]_CACHE.P  ; Read test location 0 and
U 1264, 0001.003C.0180.09D8.0000.1265  ;3862  RC[0B]_D  ; save in RC[0B]
U 1265, 0001.003C.0180.0800.0010.1266  ;3863  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 1266, 0000.013C.0180.0800.0000.10CA  ;3864  Z?   ; was the read data equal to 0?
U 10CA, 0000.003C.0180.0800.0000.10CC  ;3865 =0 J/T.44.S2.BAD.DATA ; No, Call an error
U 10CB, 0000.003C.0180.0800.0000.10CE  ;3866  J/T.44.S3  ; Yes, continue with the test
```

```
            ;3867  .PAGE
            ;3868  ;
            ;3869  ;////////////////////////////////////////////////////////////////////////////
            ;3870  ;
            ;3871  ; Data read from test location was not equal to 0, byte 2 did not get
            ;3872  ; corrected.  Call an error.
            ;3873  ;
            ;3874  =0
            ;3875  T.44.S2.BAD.DATA:
U 10CC, 0000,003D,D580,0800,0084,705C  ;3876  ERROR2[.6]
U 10CD, 0000,003C,0180,0800,0000,10CE  ;3877  NOP
            ;3878  ;
            ;3879  ;////////////////////////////////////////////////////////////////////////////
            ;3880  ;
            ;3881  ; ERROR LOGOUT:
            ;3882  ;
            ;3883  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3884  ;  I/O Address of Configuration Register C/D
            ;3885  ;  Expected Memory Data
            ;3886  ;  Received Memory Data
            ;3887  ;  Data Pattern to Force the Error
            ;3888  ;  Unused
            ;3889  ;
            ;3890  ;////////////////////////////////////////////////////////////////////////////
            ;3891  ;
            ;3892  =0
            ;3893  T.44.S3:
U 10CE, 0000,003D,0180,0800,0000,119D  ;3894  SUBTEST   ; Update the subtest counter
U 10CF, 0000,003C,0180,0800,0084,7267  ;3895  SC_K[.8]  ; Get ready to assert bit 8 in RC[0A]
U 1267, 0003,001C,0180,09D0,0000,1268  ;3896  RC[0A]_NOT.MASK  ; Load RC[0A] with x00000100 for forcing
            ;3897      ; an error to the test location
U 1268, 0003,003C,0180,09E0,0000,10D0  ;3898  RC[0C]_0  ; Load RC[0C] with the expected memory data
U 10D0, 0000,003D,0180,0800,0000,1195  ;3899 =0 ERLOOP   ; loop on error from here
U 10D1, 0000,003C,0180,0800,0000,10D2  ;3900  NOP
U 10D2, 0000,003D,0180,0800,0000,1084  ;3901 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10D3, 0018,0038,1980,0800,0200,1269  ;3902  VA_K[ZERO]  ; Point VA to test location 0
U 1269, 0810,0038,0180,0950,0000,126A  ;3903  D_RC[0A]  ; Load D with init pattern for write
U 126A, 0000,003C,0180,A800,0000,10D4  ;3904  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 10D4, 0818,0039,0980,0800,0000,11B0  ;3905 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 10D5, 0000,003C,0180,0800,0000,10D6  ;3906  NOP
U 10D6, 0818,0039,8580,0800,0000,121B  ;3907 =0 SET.ECC[.C]  ; Set substitute ECC mode
U 10D7, 0018,0038,1980,0800,0200,126B  ;3908  VA_K[ZERO]  ; Load VA with 0 for writing byte 0
U 126B, 0F00,003C,0180,0800,0000,126C  ;3909  D_0   ; data to write in byte 0
U 126C, 0000,803C,0180,A800,0000,10D8  ;3910  CACHE.P_D[BYTE]  ; Write the pattern with data type BYTE
U 10D8, 0818,0039,0D80,0800,0000,11B0  ;3911 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 10D9, 0018,0038,1980,0800,0200,126D  ;3912  VA_K[ZERO]  ; Point VA to test location 0
U 126D, 0000,003C,0180,C800,0000,126E  ;3913  D[LONG]_CACHE.P  ; Read test location 0 and
U 126E, 0001,003C,0180,09D8,0000,126F  ;3914  RC[0B]_D  ; save in RC[0B]
U 126F, 0001,003C,0180,0800,0010,1270  ;3915  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 1270, 0000,013C,0180,0800,0000,10DA  ;3916  Z?   ; was the read data equal to 0?
U 10DA, 0000,003C,0180,0800,0000,10DC  ;3917 =0 J/T.44.S3.BAD.DATA ; No, Call an error
U 10DB, 0000,003C,0180,0800,0000,10DE  ;3918  J/T.44.S4  ; Yes, continue with the test
```

```
            ;3919  .PAGE
            ;3920 ;
            ;3921 ;////////////////////////////////////////////////////////////////////////////
            ;3922 ;
            ;3923 ; Data read from test location was not equal to 0, byte 1 did not get
            ;3924 ; corrected.  Call an error.
            ;3925 ;
            ;3926 =0
            ;3927 T.44.S3.BAD.DATA:
U 10DC, 0000,003D,D580,0800,0084,705C  ;3928  ERROR2[.6]
U 10DD, 0000,003C,0180,0800,0000,10DE  ;3929  NOP
            ;3930 ;
            ;3931 ;////////////////////////////////////////////////////////////////////////////
            ;3932 ;
            ;3933 ; ERROR LOGOUT:
            ;3934 ;
            ;3935 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3936 ;  I/O Address of Configuration Register C/D
            ;3937 ;  Expected Memory Data
            ;3938 ;  Received Memory Data
            ;3939 ;  Data Pattern to Force the Error
            ;3940 ;  Unused
            ;3941 ;
            ;3942 ;////////////////////////////////////////////////////////////////////////////
            ;3943 ;
            ;3944 =0
            ;3945 T.44.S4:
U 10DE, 0000,003D,0180,0800,0000,119D  ;3946  SUBTEST   ; Update the subtest counter
U 10DF, 0018,0038,0580,09D0,0000,1271  ;3947  RC[0A]_K[.1]  ; Load RC[0A] with x00000001 for forcing
            ;3948      ; an error to the test location
U 1271, 0003,003C,0180,09E0,0000,10E0  ;3949  RC[0C]_0  ; Load RC[0C] with the expected memory data
U 10E0, 0000,003D,0180,0800,0000,1195  ;3950 =0 ERLOOP   ; loop on error from here
U 10E1, 0000,003C,0180,0800,0000,10E2  ;3951  NOP
U 10E2, 0000,003D,0180,0800,0000,1084  ;3952 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10E3, 0018,0038,1980,0800,0200,1272  ;3953  VA_K[ZERO]  ; Point VA to test location 0
U 1272, 0810,0038,0180,0950,0000,1273  ;3954  D_RC[0A]  ; Load D with init pattern for write
U 1273, 0000,003C,0180,A800,0000,10E4  ;3955  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 10E4, 0000,003D,0580,0800,0084,71D8  ;3956 =0 SET.TRAP.MASK[.1] ; Enable Unaligned data micro trap
U 10E5, 0000,003C,0180,0800,0000,10E6  ;3957  NOP
U 10E6, 0818,0039,0980,0800,0000,11B0  ;3958 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 10E7, 0000,003C,0180,0800,0000,10E8  ;3959  NOP
U 10E8, 0818,0039,8580,0800,0000,121B  ;3960 =0 SET.ECC[.C]  ; Set substitute ECC mode
U 10E9, 0018,0038,0580,0800,0200,1274  ;3961  VA_K[.1]  ; Load VA with 1 for writing bytes 1, 2, and 3
U 1274, 0F00,003C,0180,0800,0000,1275  ;3962  D_0   ; data to write in bytes 1, 2, and 3
U 1275, 0000,003C,0180,A800,0000,10EA  ;3963  CACHE.P_D[LONG]  ; Write the pattern with data type LONG
U 10EA, 0818,0039,0D80,0800,0000,11B0  ;3964 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 10EB, 0018,0038,1980,0800,0200,1276  ;3965  VA_K[ZERO]  ; Point VA to test location 0
U 1276, 0000,003C,0180,C800,0000,1277  ;3966  D[LONG]_CACHE.P  ; Read test location 0 and
U 1277, 0001,003C,0180,09D8,0000,1278  ;3967  RC[0B]_D  ; save in RC[0B]
U 1278, 0001,003C,0180,0800,0010,1279  ;3968  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 1279, 0000,013C,0180,0800,0000,10EC  ;3969  Z?   ; was the read data equal to 0?
U 10EC, 0000,003C,0180,0800,0000,10EE  ;3970 =0 J/T.44.S4.BAD.DATA ; No, Call an error
U 10ED, 0000,003C,0180,0800,0000,10F0  ;3971  J/T.44.S5  ; Yes, continue with the test
```

```
            ;3972  .PAGE
            ;3973 ;
            ;3974 ;////////////////////////////////////////////////////////////////////////////////
            ;3975 ;
            ;3976 ; Data read from test location was not equal to 0, byte 0 did not get
            ;3977 ; corrected.  Call an error.
            ;3978 ;
            ;3979 =0
            ;3980 T.44.S4.BAD.DATA:
U 10EE. 0000.003D.D580.0800.0084.705C  ;3981  ERROR2[.6]
U 10EF. 0000.003C.0180.0800.0000.10F0  ;3982  NOP
            ;3983 ;
            ;3984 ;////////////////////////////////////////////////////////////////////////////////
            ;3985 ;
            ;3986 ; ERROR LOGOUT:
            ;3987 ;
            ;3988 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3989 ;  I/O Address of Configuration Register C/D
            ;3990 ;  Expected Memory Data
            ;3991 ;  Received Memory Data
            ;3992 ;  Data Pattern to Force the Error
            ;3993 ;  Unused
            ;3994 ;
            ;3995 ;////////////////////////////////////////////////////////////////////////////////
            ;3996 ;
            ;3997 =0
            ;3998 T.44.S5:
U 10F0. 0000.003D.0180.0800.0000.119D  ;3999  SUBTEST   ; Update the subtest counter
U 10F1. 0003.003C.0180.09D0.0000.127A  ;4000  RC[0A]_0  ; Load RC[0A] with initial data pattern
U 127A. 0003.003C.0180.09E0.0000.127B  ;4001  RC[0C]_0  ; Expected data pattern
U 127B. 0018.0038.8980.09C8.0000.10F2  ;4002  RC[09]_K[.D]  ; Substitute ECC bits
U 10F2. 0000.003D.0180.0800.0000.1195  ;4003 =0 ERLOOP   ; loop on error from here
U 10F3. 0000.003C.0180.0800.0000.10F4  ;4004  NOP
U 10F4. 0000.003D.0180.0800.0000.1084  ;4005 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10F5. 0018.0038.1980.0800.0200.127C  ;4006  VA_K[ZERO]  ; Point VA to test location 0
U 127C. 0810.0038.0180.0950.0000.127D  ;4007  D_RC[0A]  ; Load D with init pattern for write
U 127D. 0000.003C.0180.A800.0000.10F6  ;4008  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 10F6. 0818.0039.0980.0800.0000.11B0  ;4009 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 10F7. 0000.003C.0180.0800.0000.10F8  ;4010  NOP
U 10F8. 0818.0039.8980.0800.0000.121B  ;4011 =0 SET.ECC[.D]  ; Set substitute ECC mode to xD
U 10F9. 0018.0038.0980.0800.0200.127E  ;4012  VA_K[.2]  ; Load VA with 2 for writing bytes 0 and 1
U 127E. 0F00.003C.0180.0800.0000.127F  ;4013  D_0   ; data to write in bytes 0 and 1
U 127F. 0000.403C.0180.A800.0000.10FA  ;4014  CACHE.P_D[WORD]  ; Write the pattern with data type WORD
U 10FA. 0818.0039.0D80.0800.0000.11B0  ;4015 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 10FB. 0018.0038.1980.0800.0200.1280  ;4016  VA_K[ZERO]  ; Point VA to test location 0
U 1280. 0000.003C.0180.C800.0000.1281  ;4017  D[LONG]_CACHE.P  ; Read test location 0 and
U 1281. 0001.003C.0180.09D8.0000.1282  ;4018  RC[0B]_D  ; save in RC[0B]
U 1282. 0001.003C.0180.0800.0010.1283  ;4019  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 1283. 0000.013C.0180.0800.0000.10FC  ;4020  Z?   ; was the read data equal to 0?
U 10FC. 0000.003C.0180.0800.0000.10FE  ;4021 =0 J/T.44.S5.BAD.DATA ; No, Call an error
U 10FD. 0000.003C.0180.0800.0000.110A  ;4022  J/T.44.S6  ; Yes, continue with the test
```

```
            ;4023  .PAGE
            ;4024 ;
            ;4025 ;////////////////////////////////////////////////////////////////////////////
            ;4026 ;
            ;4027 ; Data was modified on the array with a SBE in the check bits.  Call an error.
            ;4028 ;
            ;4029 =0
            ;4030 T.44.S5.BAD.DATA:
U 10FE. 0000.003D.D580.0800.0084.705C  ;4031  ERROR2[.6]
U 10FF. 0000.003C.0180.0800.0000.110A  ;4032  NOP
            ;4033 ;
            ;4034 ;////////////////////////////////////////////////////////////////////////////
            ;4035 ;
            ;4036 ; ERROR LOGOUT:
            ;4037 ;
            ;4038 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4039 ;  I/O Address of Configuration Register C/D
            ;4040 ;  Expected Memory Data
            ;4041 ;  Received Memory Data
            ;4042 ;  Data Pattern to Force the Error
            ;4043 ;  Unused
            ;4044 ;
            ;4045 ;////////////////////////////////////////////////////////////////////////////
            ;4046 ;
            ;4047 =0
            ;4048 T.44.S6:
U 110A. 0000.003D.0180.0800.0000.119D  ;4049  SUBTEST   ; Update the subtest counter
U 110B. 0018.0038.0D80.09D0.0000.1284  ;4050  RC[0A]_K[.3]  ; Load RC[0A] with x3 to force an error in
            ;4051     ; byte 0 of location 0
U 1284. 0003.003C.0180.09E0.0000.1110  ;4052  RC[0C]_D  ; Load RC[0C] with the expected memory data
U 1110. 0000.003D.0180.0800.0000.1195  ;4053 =0 ERLOOP   ; loop on error from here
U 1111. 0000.003C.0180.0800.0000.1112  ;4054  NOP
U 1112. 0000.003D.0180.0800.0000.1084  ;4055 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1113. 0018.0038.1980.0800.0200.1285  ;4056  VA_K[ZERO]  ; Point VA to test location 0
U 1285. 0810.0038.0180.0950.0000.1286  ;4057  D_RC[0A]  ; Load D with init pattern for write
U 1286. 0000.003C.0180.A800.0000.1114  ;4058  CACHE.P_D[LONG]  ; Write the initialize data pattern to memory
U 1114. 0818.0039.0980.0800.0000.11B0  ;4059 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 1115. 0818.0038.3180.0800.0000.1287  ;4060  D_K[.40]  ; Load D with x40
U 1287. 0819.0014.0D80.0800.0000.1288  ;4061  D_D+K[.3]  ; Add x3 to D for constant of x43 used to set
U 1288. 0001.003C.0180.09C8.0000.1116  ;4062  RC[09]_D  ; Substitute ECC bits for data of x01
U 1116. 0000.003D.0180.0800.0000.121B  ;4063 =0 CALL[SET_ECC]  ; Set substitue ECC bits according to the
            ;4064     ; contents of the D register
U 1117. 0018.0038.1980.0800.0200.1289  ;4065  VA_K[ZERO]  ; Load VA with 0 for writing byte 0
U 1289. 0F00.003C.0180.0800.0000.128A  ;4066  D_0   ; data to write in byte 0
U 128A. 0000.803C.0180.A800.0000.1118  ;4067  CACHE.P_D[BYTE]  ; Write the pattern with data type BYTE
U 1118. 0818.0039.0D80.0800.0000.11B0  ;4068 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 1119. 0018.0038.1980.0800.0200.128B  ;4069  VA_K[ZERO]  ; Point VA to test location 0
U 128B. 0000.003C.0180.C800.0000.128C  ;4070  D[LONG]_CACHE.P  ; Read test location 0 and
U 128C. 0001.003C.0180.09D8.0000.128D  ;4071  RC[0B]_D  ; save in RC[0B]
U 128D. 0001.003C.0180.0800.0010.128E  ;4072  ALU_D,CLK.UBCC  ; Load ALU with read data for checking
U 128E. 0000.013C.0180.0800.0000.111A  ;4073  Z?   ; was the read data equal to 0?
U 111A. 0000.003C.0180.0800.0000.111C  ;4074 =0 J/T.44.S6.BAD.DATA ; No, Call an error
U 111B. 0000.003C.0180.0800.0000.1040  ;4075  J/RESTART.TEST.44 ; Yes, restart test for next controller
```

```
            ;4076  .PAGE
            ;4077 ;
            ;4078 ;////////////////////////////////////////////////////////////////////////////
            ;4079 ;
            ;4080 ; Data read from test location was not equal to 0, byte 0 did not get
            ;4081 ; corrected.  Call an error.
            ;4082 ;
            ;4083 =0
            ;4084 T.44.S6.BAD.DATA:
U 111C, 0000.003D.D580.0800.0084.705C  ;4085  ERROR2[.6]
U 111D, 0000.003C.0180.0800.0000.1040  ;4086  J/RESTART.TEST.44
            ;4087 ;
            ;4088 ;////////////////////////////////////////////////////////////////////////////
            ;4089 ;
            ;4090 ; ERROR LOGOUT:
            ;4091 ;
            ;4092 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4093 ;  I/O Address of Configuration Register C/D
            ;4094 ;  Expected Memory Data
            ;4095 ;  Received Memory Data
            ;4096 ;  Data Pattern to Force the Error
            ;4097 ;  Substitute ECC Bits
            ;4098 ;
            ;4099 ;////////////////////////////////////////////////////////////////////////////
            ;4100 ;
            ;4101  ;
            ;4102  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4103  ;
            ;4104 T.44.EXIT:
U 128F, 0000.003C.0180.0800.0000.111E  ;4105  NOP
            ;4106
```

```
;4107  .PAGE "TEST 1F8 DOUBLE BIT ERROR ON MASKED WRITES"
;4108 ;*************************************************************
;4109 ;+++
;4110 ;
;4111 ; Functional Description:
;4112 ; ---------------------
;4113 ;
;4114 ; This test starts by initializing location 0 with some data.  A RDS
;4115 ; error is forced during masked writes to memory.  It is expected that:
;4116 ; if the mask is not all ones (ie., not all the bytes are written to the
;4117 ; array) then the write is aborted and thus there should be no change in
;4118 ; the initial data.  If, however, the mask is all ones (a normal longword
;4119 ; write to memory) the cycle should not be aborted.
;4120 ;
;4121 ;   [Subtest 1:]
;4122 ;
;4123 ; Array location 0 is intialized to x3, the substitute ECC bits are set
;4124 ; for data of all zeroes (xC) and in Diagnostic Mode 2 a write with data
;4125 ; of 0's is done to byte 0 (of location 0).  It is expected that the
;4126 ; write is aborted and the data on the array is unchanged.  This will be
;4127 ; done only for one of the four bytes in a longword, since the function-
;4128 ; ality of the RDS detection logic is previously tested.
;4129 ;
;4130 ;   [Subtest 2:]
;4131 ;
;4132 ; This subtest is similar to the previous one.  The difference is that
;4133 ; after setting the ECC substitute bits and the diagnostic mode, a
;4134 ; longword write will be done to location 0.  It is expected that the
;4135 ; write does not abort.
;4136 ;
;4137 ;
;4138 ; Logic Description:
;4139 ; -----------------
;4140 ;  N/A
;4141 ;
;4142 ; Error Logout:
;4143 ; ------------
;4144 ;
;4145 ; See individual error reports.
;4146 ;
;4147 ; Sync Point Description:
;4148 ; ----------------------
;4149 ;  N/A
;4150 ;
;4151 ;=============================================================
;4152 ;
;4153 ; Register Map:
;4154 ; ------------
;4155 ;
;4156 ;     RA,RB     Description:
;4157 ;     -----     -----------
;4158 ;
;4159 ; Not Used
;4160 ;
;4161 ;
```

```
            ;4162 ;       RC        Description:
            ;4163 ;       --        -----------
            ;4164 ;
            ;4165 ;        E I/O Base Address of MS780-E Currently Under Test
            ;4166 ;
            ;4167 ;        D I/O Address of Configuration Register C/D
            ;4168 ;
            ;4169 ;        C Expected Memory Data
            ;4170 ;
            ;4171 ;        B Received Memory Data
            ;4172 ;
            ;4173 ;        A Initial Memory Data Pattern
            ;4174 ;
            ;4175 ;--
            ;4176 ;*************************************************************
            ;4177 =0
U 111E, 0000.003D.0180.0800.0000.1180  ;4178 T.45: NEWTST   ; Signify the start of a new test
U 111F, 0000.003C.0180.0800.0000.1044  ;4179  NOP
U 1044, 0000.003D.0180.0800.0000.108A  ;4180 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the sbi
U 1045, 0000.003C.0180.0800.0000.12A0  ;4181  J/T.45.EXIT  ; No MS780-E's to test, exit this test
U 1046, 0000.003C.0180.0800.0000.1048  ;4182  NOP
            ;4183 =
            ;4184  ;
            ;4185  ; This is the entry point in the test to configure the controller(s).
            ;4186  ; After testing the first controller, control is returned here and
            ;4187  ; the next controller (if there is one) is configured.  If there is
            ;4188  ; not another controller then the test is exited.
            ;4189  ;
            ;4190 =00
            ;4191 RESTART.TEST.45:
U 1048, 0000.003D.0180.0800.0000.106C  ;4192  CALL[START.TEST] ; Configure the Controllers
U 1049, 0000.003C.0180.0800.0000.12A0  ;4193  J/T.45.EXIT  ; No more controllers to configure, exit test
U 104A, 0018.0038.0D80.09D0.0000.1120  ;4194  RC[0A]_K[.3]  ; Load RC[0A] with initial memory data
            ;4195 =
            ;4196 ;
            ;4197 ;=========================================================================
            ;4198 ;
            ;4199 ; Subtest 1
            ;4200 ;
U 1120, 0000.003D.0180.0800.0000.119D  ;4201 =0 SUBTEST   ; Update the Subtest counter
U 1121, 0018.0038.0D80.09E0.0000.1122  ;4202  RC[0C]_K[.3]  ; Load RC[0C] with expected memory data
U 1122, 0000.003D.0180.0800.0000.1195  ;4203 =0 ERLOOP   ; Loop on error from here
U 1123, 0000.003C.0180.0800.0000.1124  ;4204  NOP
U 1124, 0000.003D.0180.0800.0000.1084  ;4205 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1125, 0018.0038.1980.0800.0200.1290  ;4206  VA_K[ZERO]  ; Point VA to location 0
U 1290, 0810.0038.0180.0950.0000.1291  ;4207  D_RC[0A]  ; Load D with initial memory data for write
U 1291, 0000.0C3C.0180.A800.0000.1126  ;4208  CACHE.P_D[LONG]  ; Write the initial memory data to locatio 0
U 1126, 0818.0039.0980.0800.0000.11B0  ;4209 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode ECC Substitute
U 1127, 0000.003C.0180.0800.0000.1128  ;4210  NOP
U 1128, 0818.0039.8580.0800.0000.121B  ;4211 =0 SET.ECC[.C]  ; Set substitute ECC bits to xC for data of 0's
U 1129, 0018.0038.1980.0800.0200.1292  ;4212  VA_K[ZERO]  ; Load VA with 0 to write to byte 0
U 1292, 0F00.003C.0180.0800.0000.1293  ;4213  D_0   ; Load D with the data to be written
U 1293, 0000.803C.0180.A800.0000.112A  ;4214  CACHE.P_D[BYTE]  ; Write the byte of 0 to location 0
U 112A, 0818.0039.0D80.0800.0000.11B0  ;4215 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass mode
U 112B, 0018.0038.1980.0800.0200.1294  ;4216  VA_K[ZERO]  ; Point VA back to test location
```

```
ZZ-ESKAR-V22  TEST 1F8 DOUBLE BIT ERROR ON MASKED WRITES                                SEQ 1822
ESKAR52.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:56                  Page  11

U 1294, 0000.003C.0180.C800.0000.1295  ;4217  D[LONG]_CACHE.P  ; Read test location 0
U 1295, 0001.003C.0180.09D8.0000.1296  ;4218  RC[0B]_D  ; Save read data in RC[0B]
          ;4219  ALU_D.XOR.RC[0C], ; Exclusive received memory data with the
U 1296, 0011.0020.0180.0960.0010.1297  ;4220  CLK.UBCC  ; expected memory data
U 1297, 0000.013C.0180.0800.0000.112C  ;4221  Z?   ; Were they equal ?
U 112C, 0000.003C.0180.0800.0000.112E  ;4222 =0 J/T.45.S1.BAD.DATA.1 ; No, call an error
U 112D, 0000.003C.0180.0800.0000.1130  ;4223  J/T.45.S2  ; Yes, continue with the test
```

```
            ;4224  .PAGE
            ;4225 ;
            ;4226 ;////////////////////////////////////////////////////////////////////////////
            ;4227 ;
            ;4228 ; Received data did not match expected data.  Masked write was not aborted
            ;4229 ; by RDS in byte 0.  Call an error.
            ;4230 ;
            ;4231 =0
            ;4232 T.45.S1.BAD.DATA.1:
U 112E, 0000.003D.D580.0800.0084.705C  ;4233  ERROR2[.6]
U 112F, 0000.003C.0180.0800.0000.1130  ;4234  NOP
            ;4235 ;
            ;4236 ;////////////////////////////////////////////////////////////////////////////
            ;4237 ;
            ;4238 ; ERROR LOGOUT:
            ;4239 ;
            ;4240 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4241 ;  I/O Address of Configuration Register C/D
            ;4242 ;  Expected Memory Data
            ;4243 ;  Received Memory Data
            ;4244 ;  Initial Memory Pattern
            ;4245 ;  Unused
            ;4246 ;
            ;4247 ;////////////////////////////////////////////////////////////////////////////
            ;4248 ;
            ;4249 ;
            ;4250 ;============================================================================
            ;4251 ;
            ;4252 ; Subtest 2
            ;4253 ;
            ;4254 =0
            ;4255 T.45.S2:
U 1130, 0000.003D.0180.0800.0000.119D  ;4256  SUBTEST   ; Update the subtest counter
U 1131, 0003.003C.0180.09E0.0000.1132  ;4257  RC[0C]_0  ; Load RC[0C] with the expected memory data
U 1132, 0000.003D.0180.0800.0000.1195  ;4258 =0 ERLOOP   ; Loop on error from here
U 1133, 0000.003C.0180.0800.0000.1134  ;4259  NOP
U 1134, 0000.003D.0180.0800.0000.1084  ;4260 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1135, 0018.0038.1980.0800.0200.1298  ;4261  VA_K[ZERO]  ; Point VA to test location 0
U 1298, 0810.0038.0180.0950.0000.1299  ;4262  D_RC[0A]  ; Load D with the initial memory data for write
U 1299, 0000.003C.0180.A800.0000.1136  ;4263  CACHE.P_D[LONG]  ; Write the initialize memory data
U 1136, 0818.0039.0980.0800.0000.11B0  ;4264 =0 SET.DIAG.MODE[.2] ; Set diagnostic mode to ECC substitute
U 1137, 0000.003C.0180.0800.0000.1138  ;4265  NOP
U 1138, 0818.0039.8580.0800.0000.121B  ;4266 =0 SET.ECC[.C]  ; Set ECC substitute bit to xC
U 1139, 0018.0038.1980.0800.0200.129A  ;4267  VA_K[ZERO]  ; Point VA to test location 0
U 129A, 0F00.003C.0180.0800.0000.129B  ;4268  D_0   ; Load D with zero for write to memory
U 129B, 0000.003C.0180.A800.0000.113A  ;4269  CACHE.P_D[LONG]  ; Preform a longword write to location 0
U 113A, 0818.0039.0D80.0800.0000.11B0  ;4270 =0 SET.DIAG.MODE[.3] ; Set diagnostic mode to ECC bypass
U 113B, 0018.0038.1980.0800.0200.129C  ;4271  VA_K[ZERO]  ; Point VA to test location 0
U 129C, 0000.003C.0180.C800.0000.129D  ;4272  D[LONG]_CACHE.P  ; Read test location 0
U 129D, 0001.003C.0180.09D8.0000.129E  ;4273  RC[0B]_D  ; Save read data in RC[0B]
            ;4274  ALU_D.XOR.RC[0C], ; Exclusive Or Received memory data with the
U 129E, 0011.0020.0180.0960.0010.129F  ;4275  CLK.UBCC  ; expected memory data
U 129F, 0000.013C.0180.0800.0000.113C  ;4276  Z?   ; Are they equal ?
U 113C, 0000.003C.0180.0800.0000.113E  ;4277 =0 J/T.45.S2.BAD.DATA.1 ; No, call an error
U 113D, 0000.003C.0180.0800.0000.1048  ;4278  J/RESTART.TEST.45 ; Yes, end of this controller test, go restart
```

```
          ;4279     ; the test for the next controller
```

```
            ;4280  .PAGE
            ;4281  ;
            ;4282  ;////////////////////////////////////////////////////////////////////////////////
            ;4283  ;
            ;4284  ; Received memory data did not match expected memory data.  Call an error.
            ;4285  ;
            ;4286  =0
            ;4287  T.45.52.BAD.DATA.1:
U 113E,  0000,003D,D580,0800,0084,705C  ;4288  ERROR2[.6]
U 113F,  0000,003C,0180,0800,0000,1048  ;4289  J/RESTART.TEST.45
            ;4290  ;
            ;4291  ;////////////////////////////////////////////////////////////////////////////////
            ;4292  ;
            ;4293  ; ERROR LOGOUT:
            ;4294  ;
            ;4295  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4296  ;  I/O Address of Configuration Register C/D
            ;4297  ;  Expected Memory Data
            ;4298  ;  Received Memory Data
            ;4299  ;  Initial Memory Pattern
            ;4300  ;  Unused
            ;4301  ;
            ;4302  ;////////////////////////////////////////////////////////////////////////////////
            ;4303  ;
            ;4304   ;
            ;4305   ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4306   ;
            ;4307  T.45.EXIT:
U 12A0,  0000,003C,0180,0800,0000,1140  ;4308  NOP
            ;4309
```

```
;4310  .PAGE "TEST 1F9 QUICK VERIFY ARRAY DATA INTEGRITY TEST"
;4311 ;*************************************************************
;4312 ;***
;4313 ;
;4314 ; Functional Description:
;4315 ; ----------------------
;4316 ;
;4317 ; This test is a Quick Verify test of the data integrity of the array.
;4318 ;
;4319 ; PLEASE NOTE:  This test may NOT find all CRD/RDS error due to "hard
;4320 ;        failures".
;4321 ;
;4322 ; The Quick verify test runs a marching pattern test on the arrays, and
;4323 ; is intended to detect possible addressing errors and interference
;4324 ; between cells within the same RAM chip.  The test will start with
;4325 ; Board 0, Bank 0, Location 0 and will move up to higher addresses test-
;4326 ; ing one Bank at a time.  When all four banks on a board have been
;4327 ; tested, the next Board will be accessed. If a Time Out micro trap
;4328 ; occurs or if the data on the array is not equal to the expected and
;4329 ;       a RDS micro trap was generated (thus indicating a possible addressing
;4330 ;       error) the test will stop with an error call.
;4331 ;
;4332 ;
;4333 ; Logic Description:
;4334 ; -----------------
;4335 ;  N/A
;4336 ;
;4337 ;
;4338 ; Error Logout:
;4339 ; ------------
;4340 ;
;4341 ; See individual error reports.
;4342 ;
;4343 ; Sync Point Description:
;4344 ; ----------------------
;4345 ;
;4346 ;  N/A
;4347 ;
;4348 ;=============================================================
;4349 ;
;4350 ; Register Map:
;4351 ; ------------
;4352 ;
;4353 ;      RA,RB      Description:
;4354 ;      -----      -----------
;4355 ;
;4356 ;        0        Memory Size (in bytes) - 1
;4357 ;
;4358 ;        1 Loop Counter
;4359 ;
;4360 ;        2 Pass Counter
;4361 ;
;4362 ;        3 Maximum Address for Array ( 64K = 10000, 256K = 40000 )
;4363 ;
;4364 ;       RC        Description:
```

```
            ;4365 ;       --       -----------
            ;4366 ;
            ;4367 ;       E I/O Base Address of MS780-E Currently Under Test
            ;4368 ;
            ;4369 ;       D I/O Address of Configuration Register C/D
            ;4370 ;
            ;4371 ;       C Running Bank Pointer within Board
            ;4372 ;
            ;4373 ;       B Bank Pointer
            ;4374 ;
            ;4375 ;       A Board Counter
            ;4376 ;
            ;4377 ;       9 Board Pointer
            ;4378 ;
            ;4379 ;       6 Initialize Pattern to Write to Memory
            ;4380 ;
            ;4381 ; 5 Expected Pattern from Memory
            ;4382 ;
            ;4383 ; 4 Received Pattern from Memory
            ;4384 ;
            ;4385 ; 3 CRD Errors in Bank 3
            ;4386 ;
            ;4387 ; 2 CRD Errors in Bank 2
            ;4388 ;
            ;4389 ; 1 CRD Errors in Bank 1
            ;4390 ;
            ;4391 ; 0 CRD Errors in Bank 0
            ;4392 ;
            ;4393 ;--
            ;4394 ;*****************************************************************
            ;4395 =0
U 1140, 0000.003D.0180.0800.0000.1180  ;4396 T.46: NEWTST   ; Signify the start of a new test
U 1141, 0000.003C.0180.0800.0000.1050  ;4397  NOP
U 1050, 0000.003D.0180.0800.0000.108A  ;4398 =00 CALL[FIND.MS780E] ; Search out all MS780-E's on the SBI
U 1051, 0000.003C.0180.0800.0000.12E6  ;4399  J/T.46.EXIT  ; No MS780-E's to test, exit this test
U 1052, 0000.003C.0180.0800.0000.1054  ;4400  NOP
            ;4401 =
            ;4402  ;
            ;4403  ; This is the restart point for the test to configure the next
            ;4404  ; controller after it is finished with the current one.  If there
            ;4405  ; are no more controllers to test then the test is exited.
            ;4406  ;
            ;4407 =00
            ;4408 RESTART.TEST.46:
U 1054, 0000.003D.0180.0800.0000.106C  ;4409  CALL[START.TEST] ; Configure the controllers
U 1055, 0000.003C.0180.0800.0000.12E6  ;4410  J/T.46.EXIT  ; No more controllers to configure, exit test
U 1056, 0000.003C.0180.0800.0000.1142  ;4411  NOP
            ;4412 =
            ;4413 =0 SC_K[.2],  ; Get value to decrement T0
U 1142, 0000.003D.0980.0800.0084.7250  ;4414   CALL[DECREMENT.T0]
U 1143, 0000.003C.0180.0800.0000.1144  ;4415  NOP
U 1144, 0818.0039.0580.0800.0000.1241  ;4416 =0 TYPE_MSG_1  ; Print message for I/O Base Address of
            ;4417      ; MS780-E/H under test
U 1145, 0810.0038.0180.0970.0000.1146  ;4418  D_RC[0E]  ; Get I/O base address of the memory under test
U 1146, 0000.003D.0180.0800.0000.1244  ;4419 =0 TYP_RC[0E]_CRLF  ; Print the Address
```

```
U 1147, 0000,003C,0180,0800,0000,1060  ;4420  NOP
U 1060, 0000,003D,0180,0800,0000,1235  ;4421 =00 CALL[64K.OR.256K] ; Find out what kind of memory arrays are
          ;4422     ; present (64K or 256K)
          ;4423  J/T.46.MIX.ERROR, ; CASE 1:  Return here when there is a mixture
U 1061, 0018,0038,0D80,09B0,0000,1148  ;4424   RC[6]_K[.3] ; of arrays on the controller or there were
          ;4425     ; no arrays found.
          ;4426  J/T.46.64K.CHIPS, ; CASE 2:  Return here when there is 64k ram
U 1062, 0018,0038,0580,09B0,0000,12A1  ;4427   RC[6]_K[.1] ; array boards present (1 mega byte array)
          ;4428  J/T.46.256K.CHIPS, ; CASE 3:  Return here when there is 256k ram
U 1063, 0018,0038,1180,09B0,0000,12A3  ;4429   RC[6]_K[.4] ; array boards present (4 mega byte array)
```

```
            ;4430  .PAGE
            ;4431  ;
            ;4432  ;////////////////////////////////////////////////////////////////////////////////
            ;4433  ;
            ;4434  ; Got an error from 64K.OR.256K subroutine.  Either there are no arrays present
            ;4435  ; or there are a mixture of 64K and 256K arrays.  Call an error.
            ;4436  ;
            ;4437  =0
            ;4438  T.46.MIX.ERROR:
U 1148, 0000,003D,D980,0800,0084,7058  ;4439  ERROR1[.9]
U 1149, 0000,003C,0180,0800,0000,12A1  ;4440  NOP
            ;4441  ;
            ;4442  ;////////////////////////////////////////////////////////////////////////////////
            ;4443  ;
            ;4444  ; ERROR LOGOUT:
            ;4445  ;
            ;4446  ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4447  ;  I/O Address of Configuration Register C/D
            ;4448  ;  NOT USED
            ;4449  ;  NOT USED
            ;4450  ;  NOT USED
            ;4451  ;  NOT USED
            ;4452  ;  NOT USED
            ;4453  ;  NOT USED
            ;4454  ;  Array size flag (3 = Mixed or missing arrays,
            ;4455  ;     1 = 64K chip arrays,
            ;4456  ;     4 = 256K chip arrays)
            ;4457  ;
            ;4458  ;////////////////////////////////////////////////////////////////////////////////
            ;4459  ;
            ;4460   ;
            ;4461   ; Come here when the array's on the controller have 64K chips which
            ;4462   ; make-up a 1 mega byte array module.
            ;4463   ;
            ;4464  T.46.64K.CHIPS:
U 12A1, 0000,003C,6580,0800,0084,72A2  ;4465  SC_K[.10]  ; Get ready to mask bit 16
            ;4466   R[03]_NOT.MASK,  ; Assert bit 16 in R[03] for value of x10000
U 12A2, 0003,001C,0180,0A98,0000,114A  ;4467  J/T.46.GET.MEMORY.SIZE
            ;4468   ;
            ;4469   ; Come here when the array's on the controller have 256K chips which
            ;4470   ; make-up a 4 mega byte array module.
            ;4471   ;
            ;4472  T.46.256K.CHIPS:
U 12A3, 0000,003C,6580,0800,0084,72A4  ;4473  SC_K[.10]  ; Load the SC with x10
U 12A4, 0000,003C,0980,0800,0084,92A5  ;4474  SC_SC+K[.2]  ; Set-up SC with x12 for masking bit 18
U 12A5, 0003,001C,0180,0A98,0000,114A  ;4475  R[03]_NOT.MASK  ; Assert bit 18 in R[03] for value of x40000
            ;4476   ;
            ;4477   ; Come here after determining which size of array(s) are on the system.
            ;4478   ; Now get the memory size.
            ;4479   ;
            ;4480  =0
            ;4481  T.46.GET.MEMORY.SIZE:
U 114A, 0000,003D,0180,0800,0000,123A  ;4482  CALL[GET.MEMORY.SIZE] ; Get the size of the memory on the controller
            ;4483       ; that is under test
U 114B, 0001,003C,0180,0A80,0000,114C  ;4484  R[0]_D   ; Save memory size in R[0]
```

```
            ;4485 =0 SC_K[.3],   ; Get value to decrement T0
U 114C, 0000.003D.0D80.0800.0084.7250  ;4486   CALL[DECREMENT.T0]
U 114D, 0810.0038.0180.0968.0000.12A6  ;4487  D_RC[0D]  ; Move Configuration Register C/D to the D reg
U 12A6, 0000.193C.0180.0800.0000.106A  ;4488  D3-0?   ; Branch on the value of the D reg. bits 3-0
U 106A, 0818.0039.0980.0800.0000.1241  ;4489 =1010 TYPE_MSG_2  ; Type message "LOWER "
U 106B, 0000.003C.0180.0800.0000.114E  ;4490  J/T.46.PRINT.MAX.ADDR ; Go print the maximum address
U 106E, 0818.0039.0D80.0800.0000.1241  ;4491  TYPE_MSG_3  ; Type message "UPPER "
U 106F, 0000.003C.0180.0800.0000.114E  ;4492  NOP
            ;4493 =0
            ;4494 T.46.PRINT.MAX.ADDR:
U 114E, 0818.0039.1180.0800.0000.1241  ;4495  TYPE_MSG_4  ; Print message "MAX ADDRSS + 1 = "
U 114F, 0800.003C.0180.0A00.0000.1150  ;4496  D_R[0]   ; Load the D register with saved memory size
U 1150, 0000.003D.0180.0800.0000.1244  ;4497 =0 TYP_RC[0E]_CRLF  ; Print the memory size
U 1151, 0000.003C.0180.0800.0000.12A7  ;4498  NOP
U 12A7, 0003.003C.0180.0998.0000.12A8  ;4499  RC[3]_0   ; Init counter for CRD errors
U 12A8, 0003.003C.0180.0A88.0000.12A9  ;4500  R[1]_0   ; Initialize the loop counter
            ;4501  ;
            ;4502  ; Come here when it is time to initialize memory.  Write 0's to all
            ;4503  ; memory locations.
            ;4504  ;
            ;4505 T.46.INIT.MEMORY:
U 12A9, 0000.003C.0180.0A08.0200.12AB  ;4506  VA_R[1]   ; Point VA to location to be initialized
U 12AB, 0F00.003C.0180.0800.0000.12AC  ;4507  D_0   ; Load the D register with 0's for init.
U 12AC, 0000.003C.0180.A800.0000.12AD  ;4508  CACHE.P_D[LONG]  ; Initialize the current location by VA
U 12AD, 0800.003C.0180.0A08.0000.12AE  ;4509  D_R[1]   ; Load D reg with the memory pointer (loop
            ;4510      ; counter incremented by 4)
U 12AE, 0819.0014.1180.0800.0000.12AF  ;4511  D_D+K[.4]  ; Update the loop counter for next location
U 12AF, 0001.003C.0180.0A88.0000.12B0  ;4512  R[1]_D   ; and save it back in R[1]
            ;4513  ALU_D.XOR.R[0],  ; Exclusive Or Updated counter with maximum
U 12B0, 000D.0020.0180.0A00.0010.12B1  ;4514  CLK.UBCC  ; memory address
U 12B1, 0000.013C.0180.0800.0000.1152  ;4515  Z?   ; Have we initialized all of memory ?
U 1152, 0000.003C.0180.0800.0000.12A9  ;4516 =0 J/T.46.INIT.MEMORY ; No, continue with initialize
U 1153, 0003.003C.0180.09C8.0000.12B2  ;4517  RC[09]_0  ; Yes, continue with test, initialize the
            ;4518      ; board pointer
U 12B2, 0003.003C.0180.09D0.0000.1154  ;4519  RC[0A]_0  ; Initialize the Board Counter
            ;4520  ;
            ;4521  ; At this point we will start testing the memory array's.  This point
            ;4522  ; will be re-entered for every array.
            ;4523  ;
            ;4524 =0
            ;4525 T.46.MEMORY.TST.LOOP:
            ;4526  SC_K[.2],   ; Decrement T0 by 2
U 1154, 0000.003D.0980.0800.0084.7250  ;4527   CALL[DECREMENT.T0]
U 1155, 0000.003C.0180.0800.0000.1156  ;4528  NOP
U 1156, 0838.0039.0B80.0800.0000.1241  ;4529 =0 TYPE_MSG_5  ; Type message "BOARD NUMBER = "
U 1157, 0810.0038.0180.0950.0000.1158  ;4530  D_RC[0A]  ; Load the D register with the board number
U 1158, 0000.003D.0180.0800.0000.1244  ;4531 =0 TYP_RC[0E]_CRLF  ; Print the board number
U 1159, 0003.003C.0180.09A8.0000.12B3  ;4532  RC[05]_0  ; Expected pattern to be read during first pass
U 12B3, 001B.001C.1980.0AB0.0000.12B4  ;4533  R[6]_NOT.K[ZERO] ; Pattern to write to memory in the first
            ;4534      ; pass (xFFFFFFFF)
U 12B4, 0018.0038.0980.0A90.0000.12B5  ;4535  R[2]_K[.2]  ; Initialize the pass counter
            ;4536 T.46.INNER.LOOP.1:
U 12B5, 0003.003C.0180.09D8.0000.12B6  ;4537  RC[0B]_0  ; Initialize the Bank pointer
            ;4538 T.46.INNER.LOOP.2:
U 12B6, 0810.0038.0180.0958.0000.12B7  ;4539  D_RC[0B]  ; Load the D register with the Bank Pointer
```

```
U 12B7. 0000.003C.01F8.0800.0000.12B8  ;4540  Q_0   ; Load Q with 0's for shifting D w/zeroes in
U 12B8. 0000.003C.0980.0800.0084.72B9  ;4541  SC_K[.2]  ; Get ready to shift D left 2 bits
U 12B9. 0D00.003C.0180.0800.0000.12BA  ;4542  D_DAL.SC  ; Shift the Bank Pointer left by 2
U 12BA. 0010.0038.01C0.0948.0000.12BB  ;4543  Q_RC[09]  ; Load board pointer into the Q register
U 12BB. 001D.0030.0180.09E0.0000.12BC  ;4544  RC[0C]_D.OR.Q  ; Inclusive Or the running bank counter with
        ;4545     ; the board pointer
U 12BC. 0800.003C.0180.0A18.0000.12BD  ;4546  D_R[3]   ; Load the D register with the max address for
        ;4547     ; one array dependent on type of RAM chip
U 12BD. 0001.003C.0180.0A88.0000.115A  ;4548  R[1]_D   ; Load loop counter with max address count
        ;4549 =0
        ;4550 T.46.INNER.LOOP.3:
U 115A. 0000.003D.8980.0800.0084.71D8  ;4551  SET.TRAP.MASK[.D] ; Enable time out micro traps
U 115B. 0000.003C.0180.0800.0000.115C  ;4552  NOP
U 115C. 0000.003D.8580.0800.0084.71D8  ;4553 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 115D. 0010.0038.0180.0960.0200.12BE  ;4554  VA_RC[0C]  ; Point VA to location under test
U 12BE. 0000.003C.0180.C900.0000.12BF  ;4555  D[LONG]_CACHE.P  ; Read the location
U 12BF. 0001.003C.0180.09A0.0000.1064  ;4556  RC[04]_D  ; and save in the D register
U 1064. 0000.003D.0180.0800.0000.11D2  ;4557 =0 CHECK.UTRAP  ; Were there any unexpected micro traps ?
U 1065. 0000.003C.0180.0800.0000.1067  ;4558  J/T.46.3.CHK.SBI.ERR ; No, continue with the test
```

```
            ;4559  .PAGE
            ;4560 ;
            ;4561 ;////////////////////////////////////////////////////////////////////////////////
            ;4562 ;
            ;4563 ; Got a RDS or Time Out micro trap on read of test location.  Call an error.
            ;4564 ;
U 1066, 0000,003D,6180,0800,0084,705C  ;4565  ERROR2[.F]
            ;4566 ;
            ;4567 ;////////////////////////////////////////////////////////////////////////////////
            ;4568 ;
            ;4569 ; ERROR LOGOUT:
            ;4570 ;
            ;4571 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4572 ;  I/O Address of Configuration Register C/D
            ;4573 ;  Running Bank Pointer within Board
            ;4574 ;  Bank Pointer
            ;4575 ;  Board Counter
            ;4576 ;  Board Pointer
            ;4577 ;  Expected Micro Trap Flags
            ;4578 ;  Received Micro Trap Flags
            ;4579 ;  Initialize Memory Pattern
            ;4580 ;  Expected Memory Pattern
            ;4581 ;  Unused
            ;4582 ;  CRD Errors
            ;4583 ;
            ;4584 ;////////////////////////////////////////////////////////////////////////////////
            ;4585 ;
            ;4586 T.46.3.CHK.SBI.ERR:
U 1067, 001B,0010,8980,0800,0082,1070  ;4587  SC_K[.D]+1  ; Load SC with xE to specify sbi.err bit to chk
U 1070, 0000,003D,0180,0800,0000,1231  ;4588 =00 CALL[CHECK_SBI_ERR] ; Check the SBI.ERR regsiter for CRD tag
U 1071, 0000,003C,0180,0800,0000,12C2  ;4589  J/T.46.3.CHK.DATA ; CRD not set in SBI.ERR, continue testing
U 1072, 0000,003D,0180,0800,0000,1084  ;4590 =0 CALL[SBI.INIT]  ; Clear the SBI now
            ;4591  D_RC[3],  ; Get number of CRD errors on this board
U 1073, 0810,0038,8180,0918,0000,12C0  ;4592   K[.3FF]  ; Bring up slow constant
            ;4593  ALU_D.XOR.K[.3FF], ; Do we have x3FF CRD errors?
U 12C0, 0019,0020,8180,0800,0010,12C1  ;4594   CLK.UBCC ; ...
U 12C1, 0000,013C,0180,0800,0000,115E  ;4595  Z?
            ;4596 =0 J/T.46.NOT.MAX.CRD1, ; Do not have maximum number of CRDs yet
U 115E, 0019,0014,0580,0998,0000,12C2  ;4597   RC[3]_D+K[.1] ; ...increment CRD Counter
U 115F, 0000,003C,0180,0800,0000,1176  ;4598  J/T.46.MAX.CRD  ; Found maximum number of CRDs, skip board
            ;4599
            ;4600 T.46.NOT.MAX.CRD1:
            ;4601 T.46.3.CHK.DATA:
U 12C2, 0810,0038,0180,0920,0000,12C3  ;4602  D_RC[04]  ; Load saved read data into the D register
            ;4603  ALU_D.XOR.RC[05], ; Exclusive Or the Received data with the
U 12C3, 0011,0020,0180,0928,0010,12C4  ;4604  CLK.UBCC  ; expected data
U 12C4, 0000,013C,0180,0800,0000,1160  ;4605  Z?   ; Are they equal ?
U 1160, 0000,003C,0180,0800,0000,1162  ;4606 =0 J/T.46.3.BAD.DATA ; No, call an error
U 1161, 0000,003C,0180,0800,0000,1163  ;4607  J/T.46.3.DO.WRITE ; Yes, continue with the test
```

```
            ;4608  .PAGE
            ;4609 ;
            ;4610 ;////////////////////////////////////////////////////////////////////////////////
            ;4611 ;
            ;4612 ; The Received memory data did not match the Expected memory data.
            ;4613 ; Call an error.
            ;4614 ;
            ;4615 =0
            ;4616 T.46.3.BAD.DATA:
U 1162, 0000,003D,6180,0800,0084,705C  ;4617  ERROR2[.F]
            ;4618 ;
            ;4619 ;////////////////////////////////////////////////////////////////////////////////
            ;4620 ;
            ;4621 ; ERROR LOGOUT:
            ;4622 ;
            ;4623 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4624 ;  I/O Address of Configuration Register C/D
            ;4625 ;  Running Bank Pointer within Board
            ;4626 ;  Bank Pointer
            ;4627 ;  Board Counter
            ;4628 ;  Board Pointer
            ;4629 ;  Unused
            ;4630 ;  Unused
            ;4631 ;  Initialize Memory Pattern
            ;4632 ;  Expected Memory Pattern
            ;4633 ;  Received Memory Pattern
            ;4634 ;  CRD Errors on this board
            ;4635 ;
            ;4636 ;////////////////////////////////////////////////////////////////////////////////
            ;4637 ;
            ;4638 T.46.3.DO.WRITE:
U 1163, 0010,0038,0180,0960,0200,12C5  ;4639  VA_RC[0C]  ; Point VA to the location under test
U 12C5, 0800,003C,0180,0A30,0000,12C6  ;4640  D_R[6]   ; Load the D register with the test pattern
U 12C6, 0000,003C,0180,A800,0000,1164  ;4641  CACHE.P_D[LONG]  ; Write the test pattern to the test location
U 1164, 0000,003D,8980,0800,0084,71D8  ;4642 =0 SET.TRAP.MASK[.D] ; Enable time out micro traps
U 1165, 0000,003C,0180,0800,0000,1166  ;4643  NOP
U 1166, 0000,003D,8580,0800,0084,71D8  ;4644 =0 SET.TRAP.MASK[.C] ; Enable RDS micro traps
U 1167, 0000,003C,0180,C800,0000,12C7  ;4645  D[LONG]_CACHE.P  ; Read the test location and
U 12C7, 0001,003C,0180,09A0,0000,1074  ;4646  RC[04]_D  ; save the read data in RC[04]
U 1074, 0000,003D,0180,0800,0000,11D2  ;4647 =00 CHECK.UTRAP  ; Was there a RDS or time out micro trap ?
U 1075, 0000,003C,0180,0800,0000,1077  ;4648  J/T.46.3.CHK.SBI.ERR.1 ; No, continue with the test
```

```
            ;4649   .PAGE
            ;4650 ;
            ;4651 ;////////////////////////////////////////////////////////////////////////////////
            ;4652 ;
            ;4653 ; Got a RDS or Time Out micro trap on the read of the memory test location.
            ;4654 ; Call an error.
            ;4655 ;
U 1076, 0000,003D,6180,0800,0084,705C  ;4656  ERROR2[.F]
            ;4657 ;
            ;4658 ;////////////////////////////////////////////////////////////////////////////////
            ;4659 ;
            ;4660 ; ERROR LOGOUT:
            ;4661 ;
            ;4662 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4663 ;  I/O Address of Configuration Register C/D
            ;4664 ;  Running Bank Pointer within Board
            ;4665 ;  Bank Pointer
            ;4666 ;  Board Counter
            ;4667 ;  Board Pointer
            ;4668 ;  Expected Micro Trap Flags
            ;4669 ;  Received Micro Trap Flags
            ;4670 ;  Initialize Memory Pattern
            ;4671 ;  Expected Memory Pattern (Not valid on a Time-Out Utrap)
            ;4672 ;  Received Memory Pattern (Not valid on a Time-Out Utrap)
            ;4673 ;  CRD Errors on this board
            ;4674 ;
            ;4675 ;////////////////////////////////////////////////////////////////////////////////
            ;4676 ;
            ;4677 T.46.3.CHK.SBI.ERR.1:
U 1077, 001B,0010,8980,0800,0082,1078  ;4678  SC_K[.D]+1  ; Load the SC with xE to check sbi error
            ;4679      ; ...register bit d14
U 1078, 0000,003D,0180,0800,0000,1231  ;4680 =00 CALL[CHECK_SBI_ERR] ; Was the CRD bit set in the SBI error reg. ?
U 1079, 0000,003C,0180,0800,0000,12CA  ;4681  J/T.46.3.CHK.DATA.1 ; No, continue with the test
U 107A, 0000,003D,0180,0800,0000,1084  ;4682 =0 CALL[SBI.INIT]  ; Clear the SBI
            ;4683  D_RC[3],  ; Get CRD Counter
U 107B, 0810,0038,8180,0918,0000,12C8  ;4684   K[.3FF]  ; Bring up slow constant
            ;4685  ALU_D.XOR.K[.3FF], ; Do we have maximum number of CRD
U 12C8, 0019,0020,8180,0800,0010,12C9  ;4686   CLK.UBCC ; ...errors yet?
U 12C9, 0000,013C,0180,0800,0000,1168  ;4687  Z?
            ;4688 =0 J/T.46.NOT.MAX.CRD2, ; Not maximum number of CRD errors yet
U 1168, 0019,0014,0580,0998,0000,12CA  ;4689   RC[3]_D+K[.1] ; Increment CRD Counter
U 1169, 0000,003C,0180,0800,0000,1176  ;4690  J/T.46.MAX.CRD  ; Found maximum number of CRDs, skip board
            ;4691
            ;4692 T.46.NOT.MAX.CRD2:
            ;4693 T.46.3.CHK.DATA.1:
U 12CA, 0810,0038,0180,0920,0000,12CB  ;4694  D_RC[04]  ; Load the D reg. with the saved received data
            ;4695  ALU_D.XOR.R[6],  ; Exclusive Or the Received memory data with
U 12CB, 000D,0020,0180,0A30,0010,12CC  ;4696  CLK.UBCC  ; the expected memory data
U 12CC, 0000,013C,0180,0800,0000,116A  ;4697  Z?   ; Were they equal ?
U 116A, 0000,003C,0180,0800,0000,116C  ;4698 =0 J/T.46.3.BAD.DATA.1 ; No, Call an error
U 116B, 0000,003C,0180,0800,0000,116D  ;4699  J/T.46.3.DO.NXT.BANK ; Yes, continue with the test
```

```
            ;4700   .PAGE
            ;4701 ;
            ;4702 ;////////////////////////////////////////////////////////////////////////////////
            ;4703 ;
            ;4704 ; The received memory data did not match the expected memory data.
            ;4705 ; Call an error.
            ;4706 ;
            ;4707 =0
            ;4708 T.46.3.BAD.DATA.1:
U 116C, 0000,003D,6180,0800,0084,705C  ;4709  ERROR2[.F]
            ;4710 ;
            ;4711 ;////////////////////////////////////////////////////////////////////////////////
            ;4712 ;
            ;4713 ; ERROR LOGOUT:
            ;4714 ;
            ;4715 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4716 ;  I/O Address of Configuration Register C/D
            ;4717 ;  Running Bank Pointer within Board
            ;4718 ;  Bank Pointer
            ;4719 ;  Board Counter
            ;4720 ;  Board Pointer
            ;4721 ;  Unused
            ;4722 ;  Unused
            ;4723 ;  Initialize Memory Pattern
            ;4724 ;  Expected Memory Pattern
            ;4725 ;  Received Memory Pattern
            ;4726 ;  CRD Errors on this board
            ;4727 ;
            ;4728 ;////////////////////////////////////////////////////////////////////////////////
            ;4729 ;
            ;4730 T.46.3.DO.NXT.BANK:
U 116D, 0810,0038,0180,0960,0000,12CD  ;4731  D_RC[0C]  ; Load the D register with running bank pointer
U 12CD, 0019,0014,6580,09E0,0000,12CE  ;4732  RC[0C]_D+K[.10]  ; Point to the next address within this bank
U 12CE, 0800,003C,0180,0A08,0000,12CF  ;4733  D_R[1]   ; Load the D register with loop counter
            ;4734  ALU_D-K[.1],  ; Decrement the loop counter
            ;4735  R[1]_ALU,  ; restore back in R[1]
U 12CF, 0019,0000,0580,0A88,0010,12D0  ;4736  CLK.UBCC  ;
U 12D0, 0000,013C,0180,0800,0000,116E  ;4737  Z?   ; Is the loop counter equal to 0 ?
U 116E, 0000,003C,0180,0800,0000,115A  ;4738 =0 J/T.46.INNER.LOOP.3 ; No, continue with this loop
U 116F, 0000,003C,0180,0800,0000,1170  ;4739  nop
            ;4740 =0 D_K[.8],  ; Reset the timeout counter in MICMON
U 1170, 0818,0039,0180,0800,0000,105E  ;4741  CALL[CALLCON]  ; Call the monitor
U 1171, 0810,0038,0180,0958,0000,12D1  ;4742  D_RC[0B]  ; Move the bank pointer to the D register
            ;4743  ALU_D+K[.1],  ; Increment the Bank pointer then
U 12D1, 0019,0014,0580,09D8,0000,12D2  ;4744  RC[0B]_ALU  ; restore back in RC[0B]
U 12D2, 0810,0038,0180,0958,0000,12D3  ;4745  D_RC[0B]  ; Load the D register backup with bank pointer
            ;4746  ALU_D.XOR.K[.3], ; Exclusive Or the Bank pointer with x3
U 12D3, 0019,0020,0D80,0800,0010,12D4  ;4747  CLK.UBCC  ;
U 12D4, 0000,013C,0180,0800,0000,1172  ;4748  Z?   ; Is the bank pointer equal to x3 ?
U 1172, 0000,003C,0180,0800,0000,12B6  ;4749 =0 J/T.46.INNER.LOOP.2 ; No, continue with the test
U 1173, 0010,0038,01C0,0928,0000,12D5  ;4750  Q_RC[05]  ; Load the Q register with expected data and
U 12D5, 0001,2028,0180,09A8,0000,12D6  ;4751  RC[05]_NOT.Q  ; then complement it and restore back in RC[05]
U 12D6, 0000,003C,01C0,0A30,0000,12D7  ;4752  Q_R[6]   ; Load the Q register with initialize data and
U 12D7, 0001,2028,0180,0AB0,0000,12D8  ;4753  R[6]_NOT.Q  ; then complement it and restore back in R[6]
U 12D8, 0800,003C,0180,0A10,0000,12D9  ;4754  D_R[2]   ; Load the D register with the pass count
```

```
            ;4755  ALU_D-K[.1],  ; Decrement the pass count and
U 12D9, 0019,0000,0580,0A90,0010,12DA  ;4756  R[2]_ALU,CLK.UBCC ; then restore it back in R[2]
U 12DA, 0000,013C,0180,0800,0000,1174  ;4757  Z?    ; is the pass count equal to 0 ?
U 1174, 0000,003C,0180,0800,0000,12B5  ;4758 =0 J/T.46.INNER.LOOP.1 ; No, continue with the test
U 1175, 0000,003C,0180,0800,0000,1176  ;4759  NOP
            ;4760 ;
            ;4761 ;++
            ;4762 ;
            ;4763 ; This is an entry point for the following conditions:
            ;4764 ;
            ;4765 ; - x3FF CRD errors were found
            ;4766 ;
            ;4767 ; - the board was checked out and LESS than x3FF
            ;4768 ;   CRD errors were found
            ;4769 ;
            ;4770 ;
            ;4771 ;--
            ;4772 ;
            ;4773 =0
            ;4774 T.46.MAX.CRD:
            ;4775  SC_K[.2],  ; Decrement Type-Out flag by 2
U 1176, 0000,003D,0980,0800,0084,7250  ;4776   CALL[DECREMENT.TO]
U 1177, 0000,003C,0180,0800,0000,1178  ;4777  NOP
U 1178, 0818,0039,D580,0800,0000,1241  ;4778 =0 TYPE_MSG_6  ; Print "# CRD ERRORS = " message
U 1179, 0810,0038,0180,0918,0000,12DB  ;4779  D_RC[3]   ; Load the D register with the number of CRD
            ;4780      ; errors for this board
U 12DB, 0003,003C,0180,0998,0000,117A  ;4781  RC[3]_0   ; Clear CRD Counter
U 117A, 0000,003D,0180,0800,0000,1244  ;4782 =0 TYP_RC[0E]_CRLF  ; Print the number of CRD errors
            ;4783  D_R[3],   ; Load D with the max array address
U 117B, 0800,003C,01F8,0A18,0000,12DC  ;4784   Q_0  ; ...Load Q register with 0 for shifting the
            ;4785      ; ...D register
U 12DC, 0000,003C,1180,0800,0084,72DD  ;4786  SC_K[.4]  ; Load SC with 4 for left shift of 4 which is
            ;4787      ; equal to multiply by x10
U 12DD, 0D00,003C,0180,0800,0000,12DE  ;4788  D_DAL.SC  ; shift the D register left by 4 w/zeroes in
U 12DE, 0010,0038,01C0,0948,0000,12DF  ;4789  Q_RC[09]  ; Load Board pointer into the D register
U 12DF, 081D,0014,0180,0800,0000,12E0  ;4790  D_D+Q   ; Add board pointer to max array size
U 12E0, 0001,003C,0180,09C8,0000,12E1  ;4791  RC[09]_D  ; Point to location 0 of the next array
U 12E1, 0810,0038,0180,0950,0000,12E2  ;4792  D_RC[0A]  ; Load Board counter into the D register
U 12E2, 0019,0014,0580,09D0,0000,12E3  ;4793  RC[0A]_D+K[.1]  ; Increment the Board counter
U 12E3, 0810,0038,0180,0948,0000,12E4  ;4794  D_RC[09]  ; Load the D register with the Board pointer
            ;4795  ALU_D.XOR.R[0],  ; Exclusive Or the Board pointer with the
U 12E4, 000D,0020,0180,0A00,0010,12E5  ;4796  CLK.UBCC  ; maximum memory address + 1
U 12E5, 0000,013C,0180,0800,0000,117C  ;4797  Z?    ; Are they equal ?
U 117C, 0000,003C,0180,0800,0000,1154  ;4798 =0 J/T.46.MEMORY.TST.LOOP ; No, continue with this loop
U 117D, 0000,003C,0180,0800,0000,1054  ;4799  J/RESTART.TEST.46 ; Yes, restart this test for the next
            ;4800      ; controller
            ;4801  ;
            ;4802  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!
            ;4803  ;
            ;4804 T.46.EXIT:
U 12E6, 0000,003C,0180,0800,0000,117E  ;4805  NOP
            ;4806
            ;4807
```

```
            ;4808   .PAGE "TEST 1FA RESERVED"
            ;4809 =0
U 117E, 0000,003D,0180,0800,0000,1180  ;4810 T1FA: NEWTST
U 117F, 0000,003C,0180,0800,0000,12E7  ;4811   NOP
            ;4812
```

```
            ;4813  .PAGE "DUMMY NEWTST"
U 12E7, 0000,003D,0180,0800,0000,1180  ;4814 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 1839
ESKAR52.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:56           Page  13
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1929: 1908 1917= 1918= 2766= 2767= 2768= 2771=
U 1008  2363= 2365= 2367= 2372= 2352= 2353= 2354= 2355=
U 1010  2386= 2388= 2389= 2393= 2409= 2410= 2411= 2419=
U 1018  2893= 2894= 2895= 2896= 2897= 2898= 2899= 2900=
U 1020  2421= 2423= 2425= 2426= 2469= 2470= 2471= 2473=
U 1028  2517= 2518= 2519= 2521= 3270= 3271= 3272= 3273=
U 1030  3318= 3319= 3326= 3328= 1919= 1920= 1909 3601=
U 1038  3772= 3773= 3774= 1910 1972= 1974= 1911 3605=
U 1040  3783= 3784= 3785= 1913 4180= 4181= 4182= 1914
U 1048  4192= 4193= 4194= 1921 1985= 1986= 2028: 1922
U 1050  4398= 4399= 4400= 1923 4409= 4410= 4411= 1924
U 1058  2046= 2047= 2074: 1925 2092= 2093= 2108: 1926
U 1060  4421= 4424= 4427= 4429= 4557= 4558= 4565= 4587=
U 1068  2215= 2216= 4489= 4490= 2341= 2342= 4491= 4492=
U 1070  4588= 4589= 4590= 4592= 4647= 4648= 4656= 4678=
U 1078  4680= 4681= 4682= 4684= 2344= 2349= 2579= 2584=
U 1080  2589= 2594= 2599= 2600= 2646= 2647= 2726= 2727=
U 1088  2805= 2806= 2872= 2873= 2881= 2882= 2883= 2885=
U 1090  2888= 2889= 2930= 2931= 2932= 2933= 2940= 2941=
U 1098  2947= 2949= 2952= 2953= 2972= 2973= 2975= 2976=
U 10A0  3067= 3068= 3071= 3072= 3083= 3084= 3087= 3088=
U 10A8  3218= 3219= 3546= 3547= 3770= 3771= 3789= 3790=
U 10B0  3794= 3795= 3796= 3797= 3800= 3801= 3802= 3803=
U 10B8  3807= 3808= 3813= 3814= 3824= 3825= 3842= 3843=
U 10C0  3847= 3848= 3849= 3850= 3853= 3854= 3855= 3856=
U 10C8  3859= 3860= 3865= 3866= 3876= 3877= 3894= 3895=
U 10D0  3899= 3900= 3901= 3902= 3905= 3906= 3907= 3908=
U 10D8  3911= 3912= 3917= 3918= 3928= 3929= 3946= 3947=
U 10E0  3950= 3951= 3952= 3953= 3956= 3957= 3958= 3959=
U 10E8  3960= 3961= 3964= 3965= 3970= 3971= 3981= 3982=
U 10F0  3999= 4000= 4003= 4004= 4005= 4006= 4009= 4010=
U 10F8  4011= 4012= 4015= 4016= 4021= 4022= 4031= 4032=
U 1100  1894: 1895: 1896: 1897: 1898: 1899: 1900: 1901:
U 1108  1902: 1927 4049= 4050= 1903: 1904: 1905: 1906:
U 1110  4053= 4054= 4055= 4056= 4059= 4060= 4063= 4065=
U 1118  4068= 4069= 4074= 4075= 4085= 4086= 4178= 4179=
U 1120  4201= 4202= 4203= 4204= 4205= 4206= 4209= 4210=
U 1128  4211= 4212= 4215= 4216= 4222= 4223= 4233= 4234=
U 1130  4256= 4257= 4258= 4259= 4260= 4261= 4264= 4265=
U 1138  4266= 4267= 4270= 4271= 4277= 4278= 4288= 4289=
U 1140  4396= 4397= 4414= 4415= 4416= 4418= 4419= 4420=
U 1148  4439= 4440= 4482= 4484= 4486= 4487= 4495= 4496=
U 1150  4497= 4498= 4516= 4517= 4527= 4528= 4529= 4530=
U 1158  4531= 4532= 4551= 4552= 4553= 4554= 4597= 4598=
U 1160  4606= 4607= 4617= 4639= 4642= 4643= 4644= 4645=
U 1168  4689= 4690= 4698= 4699= 4709= 4731= 4738= 4739=
U 1170  4741= 4742= 4749= 4750= 4758= 4759= 4776= 4777=
U 1178  4778= 4779= 4782= 4784= 4798= 4799= 4810= 4811=
U 1180  1961 1962 1963 1964 1965 1966 1967 1968
U 1188  1969 1970 1971 1975 1976 1977 1978 1979
U 1190  1980 1981 1982 1983 1984 2003 2026 2027
U 1198  2072 2073 2156 2157 2158 2176 2178 2204
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1840
ESKARS2.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:56     Page  13

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2205 2206 2207 2208 2209 2210 2211 2213
U 11A8   2214 2350 2351 2376 2396 2397 2398 2405
U 11B0   2545 2546 2547 2548 2549 2550 2574 2623
U 11B8   2648 2649 2650 2651 2670 2671 2672 2673
U 11C0   2692 2693 2694 2695 2721 2755 2757 2758
U 11C8   2759 2761 2762 2763 2764 2765 2772 2773
U 11D0   2774 2775 2798 2799 2800 2801 2803 2804
U 11D8   2833 2834 2835 2874 2875 2876 2877 2886
U 11E0   2887 2891 2892 2905 2906 2908 2909 2910
U 11E8   2912 2917 2918 2919 2921 2926 2928 2929
U 11F0   2937 2938 2942 2943 2944 2945 2946 2954
U 11F8   2955 2956 2958 2959 2960 2961 2962 2963
U 1200   2968 2970 2971 2974 2998 2999 3000 3001
U 1208   3029 3030 3032 3033 3034 3063 3065 3066
U 1210   3070 3074 3075 3076 3077 3079 3080 3081
U 1218   3082 3085 3086 3112 3113 3114 3115 3116
U 1220   3117 3164 3166 3167 3168 3170 3172 3173
U 1228   3175 3177 3178 3179 3180 3183 3184 3185
U 1230   3188 3213 3214 3216 3217 3264 3265 3266
U 1238   3268 3269 3312 3313 3314 3315 3316 3330
U 1240   3332 3373 3416 3456 3498 3541 3542 3545
U 1248   3550 3590 3592 3593 3595 3597 3599 3607
U 1250   3645 3646 3648 3791 3793 3798 3799 3804
U 1258   3805 3809 3810 3811 3812 3844 3846 3851
U 1260   3852 3857 3858 3861 3862 3863 3864 3896
U 1268   3898 3903 3904 3909 3910 3913 3914 3915
U 1270   3916 3949 3954 3955 3962 3963 3966 3967
U 1278   3968 3969 4001 4002 4007 4008 4013 4014
U 1280   4017 4018 4019 4020 4052 4057 4058 4061
U 1288   4062 4066 4067 4070 4071 4072 4073 4105
U 1290   4207 4208 4213 4214 4217 4218 4220 4221
U 1298   4262 4263 4268 4269 4272 4273 4275 4276
U 12A0   4308 4465 4467 4473 4474 4475 4488 4499
U 12A8   4500 4506 2134: 4507 4508 4509 4511 4512
U 12B0   4514 4515 4519 4533 4535 4537 4539 4540
U 12B8   4541 4542 4543 4544 4546 4548 4555 4556
U 12C0   4594 4595 4602 4604 4605 4640 4641 4646
U 12C8   4686 4687 4694 4696 4697 4732 4733 4736
U 12D0   4737 4744 4745 4747 4748 4751 4752 4753
U 12D8   4754 4756 4757 4781 4786 4788 4789 4790
U 12E0   4791 4792 4793 4794 4796 4797 4805 4814
U 12E8 - 13EF Unused
U 13F0   2118: 2119: 2120: 2121: 2122: 2123: 2124: 2125:
U 13F8   2126: 2127: 2128: 2129: 2130: 2131: 2132: 2133:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKARS2	760
Used	760
Remaining	

Total microwords used in memory U: 760
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKARS2.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR53.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 1843

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1850 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1986   ERROR1 WITH PARAMETER
; 2011   ERROR 2 WITH PARAMETER
; 2057   TYPE ASCII MESSAGE ROUTINE
; 2098   TYPE ASCII MESSAGE WITH CRLF
; 2141   TYPE NUMERIC ROUTINE
; 2181   TYPE NUMERIC WITH CRLF
; 2223   COMMON CODE FOR TYPE NUMERIC ROUTINES
; 2275   COMMON CODE FOR TYPE ASCII AND NUMERIC
; 2332   DECREMENT TO ROUTINE
; 2373   ASCII MESSAGES
; 2437   NOINTR ROUTINE
; 2499   Wait Loop Routine
; 2645   FIND MEMORY AT TR
; 2898   OR FOR ONES ON MS780E
; 2971   RESET SUBTEST NUMBER
; 2994   Initialize the SBI
; 3022   Set Trap Mask
; 3051   Find MS780-E Memory
; 3194   Generate IO Address
; 3220   Set Starting Address
; 3254   Select Controller
; 3292   SELECT LOWER CONTROLLER
; 3341   SELECT UPPER CONTROLLER
; 3390   ENABLE NEXT MS780-E
; 3442   CONFIGURE MS780-E/H
; 3516   GET MEMORY SIZE
; 3575   64K OR 256K CHIPS
; 3628 TEST 1FB SBI/CACHE MISCELLANEOUS TESTS
; 4160 TEST 1FC ROM CHECKSUM TEST
; 4641 TEST 1FD RESERVED
; 4646 TEST 1FE RESERVED
; 4651 DUMMY NEWTST

```
:1  .NOLIST
:1804  .LIST
:1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR53
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; MODIFIED BY:
;1836 ;
;1837 ;  1.1 March 10, 1983  Paul Hakala
;1838 ;   Modified the GET.MEMORY.SIZE routine to calculate
;1839 ;   the correct memory address when 256k ram chips are
;1840 ;   present.
;1841 ;
;1842 ;
;1843 ;
;1844 ;
;1845 ;
;1846 ;*****************************************************************************
;1847 ;
```

```
                ;1848  .PAGE
                ;1849
                ;1850  .TOC " MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
                ;1851  ;*
                ;1852  ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1853  ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
                ;1854  ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1855  ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1856  ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1857  ;
                ;1858  ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1859  ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1860  ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1861  ; R[0F] AND DO A RETURN0.
                ;1862  ;
                ;1863  ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1864  ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1865  ; TO THE CONSOLE.
                ;1866  ;
                ;1867  ;-
U 1100, 0000.003C.1980.0800.0084.7003   ;1868 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7001   ;1869 1101: SC_K[.1],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7001   ;1870 1102: SC_K[.2],J/TRPH0 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7001   ;1871 1103: SC_K[.3],J/TRPH0 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7001   ;1872 1104: SC_K[.4],J/TRPH0 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7001   ;1873 1105: SC_K[.6],J/TRPH0 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7001   ;1874 1106: SC_K[.9],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7001   ;1875 1107: SC_K[.7],J/TRPH0 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7001   ;1876 1108: SC_K[.8],J/TRPH0 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7001   ;1877 110C: SC_K[.C],J/TRPH0 ; RDS
U 110D, 0000.003C.8980.0800.0084.7001   ;1878 110D: SC_K[.D],J/TRPH0 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7001   ;1879 110E: SC_K[.10],J/TRPH0 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003   ;1880 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.1006   ;1881 TRPH0: Q_ID[USTACK]
U 1006, 0C00.003C.01E0.0800.0000.100E   ;1882  Q_D,D_Q
U 100E, 0000.003C.8180.3C00.0000.1017   ;1883  ID[USTACK]_D
U 1017, 0C00.003C.01E0.0800.0000.1018   ;1884  Q_D,D_Q
U 1018, 0000.003C.01C0.0A78.0000.101A   ;1885  Q_R[0F]
U 101A, 0001.2024.0180.0800.0010.101C   ;1886  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 101C, 0000.013C.0180.0800.0000.1002   ;1887  Z?    ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000   ;1888 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1889  ;*
                ;1890  ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1891  ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1892  ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1893  ;
                ;1894  ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1895  ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1896  ;       FAULT STATUS REGISTER
                ;1897  ;       SBI ERROR REGISTER
                ;1898  ;       TIMEOUT ADDRESS REGISTER
                ;1899  ;       MAINTENANCE REGISTER
                ;1900  ;       CACHE PARITY REGISTER
                ;1901  ;       TB ERROR REGISTER 1
                ;1902  ;       TB ERROR REGISTER 0
```

```
            ;1903 ;-
U 1003, 0018.0038.1D80.09E8.0000.1004  ;1904 TRPH1: RC[0D]_SC
U 1004, 0000.003D.D980.0800.0084.71BB  ;1905 =0 SETRCF[.9]
U 1005, 0000.003C.49F0.2C00.0000.101E  ;1906  Q_ID[TBER0]
U 101E, 0001.203C.4DF0.2DB0.0000.1022  ;1907  RC[6]_Q,Q_ID[TBER1]
U 1022, 0001.203C.65F0.2DB8.0000.1026  ;1908  RC[7]_Q,Q_ID[SBI.ERR]
U 1026, 0001.203C.6DF0.2DD8.0000.102A  ;1909  RC[0B]_Q,Q_ID[FAULT]
U 102A, 0001.203C.75F0.2DE0.0000.102E  ;1910  RC[0C]_Q,Q_ID[MAINT]
U 102E, 0001.203C.79F0.2DC8.0000.1032  ;1911  RC[9]_Q,Q_ID[PARITY]
U 1032, 0001.203C.69F0.2DC0.0000.1036  ;1912  RC[8]_Q,Q_ID[TIME.ADDR]
U 1036, 0001.203C.81F0.2DD0.0000.1000  ;1913  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.119D  ;1914 1000: RC[0E]_Q,ERROR1
            ;1915 ;*
            ;1916 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1917 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1918 ;  NEWTST
            ;1919 ;  ERLOOP
            ;1920 ;  ERROR1
            ;1921 ;  ERROR2
            ;1922 ;
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "   Newtest Routine"
                ;1926 ;**
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to     0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 103A, 0000,003C,65F8,0800,0084,703E   ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 103E, 0818,0038,8D80,0800,0000,1042   ;1955  d_k[.1f]  ; D=1F
U 1042, 0D00,003C,0180,0800,0000,1046   ;1956  d_dal.sc  ; D=001F0000
U 1046, 0000,003C,3D80,3C00,0000,104A   ;1957  id[psl]_d  ; psl = 001F0000
U 104A, 0818,0038,0580,0800,0000,1053   ;1958  d_k[.1]   ; Clear the CES register
U 1053, 0D00,003C,0180,0800,0000,1057   ;1959  d_dal.sc  ; D = 00010000
U 1057, 0F00,003C,3180,3C00,0000,105B   ;1960  id[ces]_d,d_0  ; ces = 00010000
U 105B, 0000,003C,5D80,3C00,0000,105F   ;1961  id[acc.cs]_d  ; acc.cs = 0
U 105F, 0000,003C,3980,3C00,0000,1087   ;1962  id[sir]_d  ; sir = 0
U 1087, 0000,003C,2980,3C00,0000,108B   ;1963  id[clk.cs]_d  ; clk.cs = 0
U 108B, 0000,003C,4980,3C00,0000,100C   ;1964  id[tber0]_d  ; tber0 = 0
U 100C, 0000,003D,0180,0800,0000,11DA   ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1966  intrpt.strobe,  ; Strobe interrupts
U 100D, 0818,0038,C180,0800,4000,1093   ;1967       d_k[.ffff]  ; Setup to clear SBI error register and
U 1093, 0801,0028,6580,3C00,0000,1097   ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 1097, 0F00,003C,6D80,3C00,0000,109B   ;1969  id[fault]_d,d_0  ; ...
U 109B, 0000,003C,6D80,3C00,0000,109F   ;1970  id[fault]_d  ; fault = 0
U 109F, 0000,003C,7580,3C00,0000,1109   ;1971  id[maint]_d  ; MAINT = 0
U 1109, 0000,003C,6580,3C00,0000,1154   ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 1154, 0000,003C,7180,3C00,0000,1198   ;1973  id[comp]_d  ; comp reg = 0
U 1198, 0000,003C,E580,3C00,0000,1199   ;1974  id[T9]_d  ; Clear code for subroutine START.TEST
U 1199, 0001,003C,0180,09F0,0000,119A   ;1975  rc[0e]_d  ;
U 119A, 0001,003C,0180,0AF8,0000,119B   ;1976  r[0f]_d  ; Clear expected trap mask
U 119B, 0001,003C,0180,09F8,0000,1020   ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1850
ESKAR53.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:55                   Page   5

U 1020, 0000,003D,0180,0800,0000,11ED  ;1978 =0 call[disable.cache] ; Disable the cache
U 1021, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
        ;1980
U 119C, 0818,0038,0980,0800,0000,105E  ;1981 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 119D, 0810,0038,0180,0978,0000,119E  ;1982 ERROR1: D_RC[0F]
U 119E, 0819,0034,4D80,0800,0000,104E  ;1983  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;1984 104E: D_D.OR.K[.4],J/CALLCON
```

```
            ;1985  .PAGE
            ;1986  .TOC "    ERROR1 WITH PARAMETER"
            ;1987 ;++
            ;1988 ; FUNCTIONAL DESCRIPTION:
            ;1989 ;
            ;1990 ; This subroutine takes  as parameter the number of arguments
            ;1991 ; to be printed to the console in the case of an error logout.
            ;1992 ;
            ;1993 ; CALLING CONSTRAINT:
            ;1994 ;
            ;1995 ; =0
            ;1996 ;INPUTS:
            ;1997 ;
            ;1998 ; RC[OF] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;1999 ;--
            ;2000 =0
            ;2001 ERR1.ARG:
U 1024, 0000,003D,0180,0800,0000,11BB  ;2002  CALL[SETRCF]  ; Set the number of arguments in RC[OF]
U 1025, 0000,003C,0180,0800,0000,119D  ;2003  J/ERROR1  ; Go to ERROR1
            ;2004 ;
            ;2005 ;
            ;2006
U 119F, 0810,0038,0180,0978,0000,11A0  ;2007 ERROR2: D_RC[OF]
U 11A0, 0819,0034,4D80,0800,0000,105A  ;2008  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2009 105A: D_D.OR.K[.6],J/CALLCON
```

```
            ;2010   .PAGE
            ;2011   .TOC "    ERROR 2 WITH PARAMETER"
            ;2012 ;**
            ;2013 ; FUNCTIONAL DESCRIPTION:
            ;2014 ;
            ;2015 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2016 ; except that in this case after setting the number of arguments
            ;2017 ; too the console, control is transferred to routine ERROR2.
            ;2018 ;
            ;2019 ; INPUTS:
            ;2020 ;
            ;2021 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2022 ;
            ;2023 ;--
            ;2024 =0
            ;2025 ERR2.ARG:
U 1028, 0000,003D,0180,0800,0000,11BB  ;2026  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1029, 0000,003C,0180,0800,0000,119F  ;2027  J/ERROR2  ; Go to ERROR2
            ;2028 ;
            ;2029 ;
            ;2030
            ;2031 105E:
            ;2032 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2033  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2034 ;*
            ;2035 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2036 ;-
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2037 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2038 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2039 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2040 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2041 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2042 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2043 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2044 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2045 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2046 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2047 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2048 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2049 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2050 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2051 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;2052 13FF: NOP
U 1155, 0000,003C,0180,0800,0000,12AA  ;2053 1155: NOP
U 12AA, 0000,003C,0180,0800,0000,11A1  ;2054 12AA: NOP
            ;2055
```

```
;2056  .PAGE
;2057  .TOC "    TYPE ASCII MESSAGE ROUTINE"
;2058 ;*****************************************************************
;2059 ;**
;2060 ;
;2061 ; Functional Description:
;2062 ; ----------------------
;2063 ;
;2064 ; This routine is used to type ASCII messages on
;2065 ; the console terminal.
;2066 ;
;2067 ;
;2068 ; Calling Constraint: =0
;2069 ; ------------------
;2070 ;
;2071 ;
;2072 ; Inputs:
;2073 ; ------
;2074 ;
;2075 ; The D Register must hold the number of the message
;2076 ; desired to be typed.
;2077 ;
;2078 ; Outputs:
;2079 ; -------
;2080 ;
;2081 ; NONE
;2082 ;
;2083 ;
;2084 ; Side Effects:
;2085 ; ------------
;2086 ;
;2087 ; The contents of the following registers will be lost:
;2088 ;
;2089 ; D, SC
;2090 ;
;2091 ;--
;2092 ;*****************************************************************
;2093 MSGCOM:
;2094  SC_K[.A],   ; Get function code for Mic Mon
U 11A1, 0000,003C,F580,0800,0084,71AA  ;2095   J/COMMON  ; Go to the common portion of code
;2096
```

```
;2097  .PAGE
;2098  .TOC "    TYPE ASCII MESSAGE WITH CRLF"
;2099 ;*************************************************************
;2100 ;**
;2101 ;
;2102 ; Functional Description:
;2103 ; ----------------------
;2104 ;
;2105 ; This routine is used to type ASCII messages on the
;2106 ; console terminal.  The message is follwed by a Carriage
;2107 ; Return and a Line Feed.
;2108 ;
;2109 ;
;2110 ; Calling Constraint: =0
;2111 ; ------------------
;2112 ;
;2113 ;
;2114 ; Inputs:
;2115 ; ------
;2116 ;
;2117 ; The D register must hold the number of the message
;2118 ; to be printed.
;2119 ;
;2120 ; Outputs:
;2121 ; -------
;2122 ;
;2123 ; NONE
;2124 ;
;2125 ;
;2126 ; Side Effects:
;2127 ; ------------
;2128 ;
;2129 ; The contents of the follwoing registers will be lost:
;2130 ;
;2131 ; D, Q, SC
;2132 ;
;2133 ;--
;2134 ;*************************************************************
;2135 MSGCOM_CRLF:
U 11A2, 0038,0038,D9C0,0800,0000,11A3  ;2136  Q_K[.9].LEFT,SI/ZERO  ; Get Function code for Mic Mon
;2137  SC_Q,    ;...
U 11A3, 0001,203C,0180,0800,0082,11AA  ;2138   J/COMMON  ; ...Go to the common code
;2139
```

```
;2140  .PAGE
;2141  .TOC "    TYPE NUMERIC ROUTINE"
;2142 ;*************************************************************
;2143 ;++
;2144 ;
;2145 ; Functional Description:
;2146 ; ----------------------
;2147 ;
;2148 ; This subroutine is used for typing numerical data
;2149 ; on the console terminal.
;2150 ;
;2151 ; Calling Constraint: =0
;2152 ; ------------------
;2153 ;
;2154 ;
;2155 ; Inputs:
;2156 ; ------
;2157 ;
;2158 ; Register D must hold the numeric data to be typed.
;2159 ;
;2160 ;
;2161 ; Outputs:
;2162 ; -------
;2163 ;
;2164 ; NONE
;2165 ;
;2166 ;
;2167 ; Side Effects:
;2168 ; ------------
;2169 ;
;2170 ; The contents of the following registers will be lost:
;2171 ;
;2172 ; D, SC, ID[29]
;2173 ;
;2174 ;--
;2175 ;*************************************************************
;2176 RCECOM:
;2177  SC_K[.10],   ; Get Function Code for Mic Mon
U 11A4, 0000,003C,6580,0800,0084,71A6  ;2178   J/RCE.COMMON  ; ...Go to the common code
;2179
```

```
;2180  .PAGE
;2181  .TOC "   TYPE NUMERIC WITH CRLF"
;2182 ;*****************************************************************
;2183 ;++
;2184 ;
;2185 ; Functional Description:
;2186 ; ----------------------
;2187 ;
;2188 ; This routine is used to type numeric data on the
;2189 ; console terminal.  The data is followed by a Carriage
;2190 ; Return and a Line Feed.
;2191 ;
;2192 ;
;2193 ; Calling Constraint: -0
;2194 ; ------------------
;2195 ;
;2196 ;
;2197 ; Inputs:
;2198 ; ------
;2199 ;
;2200 ; The D register must hold the numeric data to be typed.
;2201 ;
;2202 ;
;2203 ; Outputs:
;2204 ; -------
;2205 ;
;2206 ; NONE
;2207 ;
;2208 ;
;2209 ; Side Effects:
;2210 ; ------------
;2211 ;
;2212 ; The contents of the following registers will be lost:
;2213 ;
;2214 ; D, SC, ID[29]
;2215 ;
;2216 ;--
;2217 ;*****************************************************************
;2218 RCECOM_CRLF:
;2219   SC_K[.14],    ; Get function code for Mic Mon
U 11A5, 0000,003C,2180,0800,0084,71A6  ;2220   J/RCE.COMMON  ; ...Go to the common code
;2221
```

```
            ;2222  .PAGE
            ;2223  .TOC "   COMMON CODE FOR TYPE NUMERIC ROUTINES"
            ;2224 ;*************************************************************
            ;2225 ;**
            ;2226 ;
            ;2227 ; Functional Description:
            ;2228 ; ----------------------
            ;2229 ;
            ;2230 ; This portion of code will be used by the routines that type
            ;2231 ; numeric data to the console terminal.
            ;2232 ; Its function is to save the contents of register RC[0E]
            ;2233 ; and replace them with the data to be typed.
            ;2234 ; The old contents of register RC[0E] will be reinstated before
            ;2235 ; the control is returned to the caller.
            ;2236 ;
            ;2237 ; Calling Constraint: =0
            ;2238 ; ------------------
            ;2239 ;
            ;2240 ;
            ;2241 ; Inputs:
            ;2242 ; ------
            ;2243 ;
            ;2244 ; As setup by RCECOM and RCECOM_CRLF ROUTINES.
            ;2245 ;
            ;2246 ;
            ;2247 ; Outputs:
            ;2248 ; -------
            ;2249 ;
            ;2250 ; NONE
            ;2251 ;
            ;2252 ;
            ;2253 ; Side Effects:
            ;2254 ; ------------
            ;2255 ;
            ;2256 ; Contents of teh following registers will be lost:
            ;2257 ;
            ;2258 ; D, Q, SC, ID[29]
            ;2259 ;
            ;2260 ;--
            ;2261 ;*************************************************************
            ;2262 RCE.COMMON:
U 11A6, 0000,003C,01E0,0800,0000,11A7  ;2263  Q_D     ; Save data to be typed in Q Register
U 11A7, 0810,0038,0180,0970,0000,11A8  ;2264  D_RC[0E]    ; Get contents of RC[0E]
            ;2265  ID[29]_D,    ; Save contents of RC[0E] in ID
            ;2266       ; ...register 29
U 11A8, 0001,203C,A580,3DF0,0000,102C  ;2267   RC[0E]_Q  ; Get data to be typed in RC[0E]
U 102C, 0000,003D,0180,0800,0000,11AA  ;2268 =0 CALL[COMMON]    ; CALL the common code
U 102D, 0000,003C,A5F0,2C00,0000,11A9  ;2269  Q_ID[29]    ; Retrieve saved contents
            ;2270       ; ...of RC[0E]
            ;2271  RC[0E]_Q,    ; Reinstate RC[0E]
U 11A9, 0001,203E,0180,09F0,0000,0001  ;2272   RETURN1    ; ...and return to caller
            ;2273
```

```
            ;2274  .PAGE
            ;2275  .TOC "    COMMON CODE FOR TYPE ASCII AND NUMERIC"
            ;2276 ;*****************************************************
            ;2277 ;**
            ;2278 ;
            ;2279 ; Functional Description:
            ;2280 ; ----------------------
            ;2281 ;
            ;2282 ; This code is used by both Type-ASCII and Type-Numeric
            ;2283 ; routines.  It provides the interface to the Micro Monitor.
            ;2284 ;
            ;2285 ;
            ;2286 ; Calling Constraint: =0
            ;2287 ; ------------------
            ;2288 ;
            ;2289 ;
            ;2290 ; Inputs:
            ;2291 ; ------
            ;2292 ;
            ;2293 ; As set up by MSGCOM, MSGCOM_CRLF, RCECOM and RCECOM_CRLF.
            ;2294 ;
            ;2295 ;
            ;2296 ; Outputs:
            ;2297 ; -------
            ;2298 ;
            ;2299 ; NONE
            ;2300 ;
            ;2301 ;
            ;2302 ; Side Effects:
            ;2303 ; ------------
            ;2304 ;
            ;2305 ; The contents of the following registers will be lost:
            ;2306 ;
            ;2307 ; D, Q, SC, ID[29]
            ;2308 ;
            ;2309 ;--
            ;2310 ;*****************************************************************
            ;2311 COMMON:
U 11AA, 0000,003C,0180,0800,0100,11AB  ;2312  FE_SC     ; Save the Function Code for Mic Mon
            ;2313  SC_K[.9],    ; Get ready to shift message number
U 11AB, 0000,003C,D9F8,0800,0084,71AC  ;2314   Q_0     ; ...(or type ASCII)
U 11AC, 0D00,003C,0180,0800,0000,11AD  ;2315  D_DAL.SC    ; Shift
            ;2316 CNFCOM:
U 11AD, 0000,003C,C1F0,2C00,0000,11AE  ;2317  Q_ID[T0]    ; Get the Type-Out flag
            ;2318  ALU_Q,CLK.UBCC,    ; See if the Type-Out flag is set.
U 11AE, 0C01,203C,01E0,0800,0010,11AF  ;2319   Q_D,D_Q    ; Get Type-Out flag to D
            ;2320  ALU.N?,     ; Is the Type-Out flag set?
U 11AF, 0819,1B14,0580,0800,0000,1007  ;2321   D_D+K[.1]  ; Clear Type-Out flag
            ;2322 =0111 RETURN1,    ; Type-Out flag was NOT set
U 1007, 0C00,003E,0180,0800,0000,0001  ;2323   D_Q    ;
            ;2324  ID[T0]_D,    ; Save Type-Out flag
            ;2325   D_Q,    ; Get number of the messge to
            ;2326       ; ...print
U 100F, 0C00,003C,C180,3C00,0081,11B0  ;2327   SC_FE    ; Get the Function Code for Mic Mon
            ;2328  D_D.OR.K[SC],    ; Get Function Code in the D Reg
```

ZZ-ESKAR-V22 MODULE REVISION HISTORY SEQ 1859
ESKAR53.MCR MICRO2 1P(00) 26-JUL-85 17:06:55 Page 6

```
U 11B0, 0819,0030,1D80,0800,0000,105E  ;2329   J/CALLCON  ; Call Mic Mon
            ;2330
```

```
            ;2331  .PAGE
            ;2332  .TOC "   DECREMENT TO ROUTINE"
            ;2333 ;*************************************************************
            ;2334 ;++
            ;2335 ;
            ;2336 ; Functional Description:
            ;2337 ; ----------------------
            ;2338 ;
            ;2339 ; This routine is used to set the Type-Out flag.
            ;2340 ;
            ;2341 ; Calling Constraint:
            ;2342 ; ------------------
            ;2343 ;
            ;2344 ;
            ;2345 ; Inputs:
            ;2346 ; ------
            ;2347 ;
            ;2348 ; SC must hold the number of messages to be
            ;2349 ;
            ;2350 ;
            ;2351 ; Outputs:
            ;2352 ; -------
            ;2353 ;
            ;2354 ; T0 is loaded with 0 - (SC)
            ;2355 ;
            ;2356 ;
            ;2357 ; Side Effects:
            ;2358 ; ------------
            ;2359 ;
            ;2360 ; The contents of the following regsiters will be lost
            ;2361 ;
            ;2362 ; D, SC, Q, T0
            ;2363 ;
            ;2364 ;--
            ;2365 ;*************************************************************
            ;2366 DECREMENT.T0:
U 11B1, 0F00.003C,0180.0800.0000.11B2  ;2367  D_0     ; Clear the D Register
U 11B2, 0819.0000,1D80.0800.0000.11B3  ;2368  D_D-K[SC]    ; Decrement by the specified amount
            ;2369  ID[T0]_D,    ; Save in T0
U 11B3, 0000.003E,C180.3C00.0000.0001  ;2370   RETURN1    ; Return to caller
            ;2371
            ;2372
            ;2373  .TOC "   ASCII MESSAGES"
            ;2374 ;*
            ;2375 ; FOLLOWING ARE THE ASCII MESSAGES USED BY THIS MODULE
            ;2376 ; NOTE: DO NOT CHANGE MESSAGE NUMBERS 1 AND 2 WITHOUT
            ;2377 ; CHANGING THE 'CONFIG' ROUTINE.
            ;2378 ;
            ;2379 ;& 01,/MS780-A  4K/
            ;2380 ;& 02,/MS780-C 16K/
            ;2381 ;& 03,/MS780-E 64K/
            ;2382 ;& 04,/MS780-H 256K/
            ;2383 ;& 05,/ CHIP AT TR /
            ;2384 ;& 06,/  M8213/
            ;2385 ;& 07,/  M8376/
```

```
            ;2386 ;& 08./  ROMS OK/
            ;2387 ;& 09./  ? NO M8213 OR M8376 ROMS/
            ;2388 ;& 0A,/DEPOSIT CTRLR ADRS IN RC0 AND TYPE LO/
            ;2389
            ;2390 ;*
            ;2391 ; THIS ROUTINE IS USED TO DECREMENT THE TYPEOUT FLAG ID[T0]
            ;2392 ;-
            ;2393 DECFLG:
U 11B4, 0000,003C,C1F0,2C00,0000,11B5  ;2394  Q_ID[T0]  ; GET THE FLAG
U 11B5, 0819,2000,0580,0800,0000,11B6  ;2395  D_Q-K[.1]  ; DECREMENT IT
U 11B6, 0000,003E,C180,3C00,0000,0001  ;2396  ID[T0]_D,RETURN1 ; ...
            ;2397 ;*
            ;2398 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2399 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2400 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2401 ; TYPING IT.
            ;2402 ;-
U 11B7, 0810,0038,4D80,0978,0000,11B8  ;2403 SUBTST: D_RC[0F],KMX/.FF00
U 11B8, 0019,4016,4D80,09F8,0000,0001  ;2404  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2405 ;*****************************************************************
            ;2406 ;*      THIS SUBROUTINE IS USED TO SET THE SUBTEST
            ;2407 ; NUMBER BACK TO 1, WHEN THERE IS A TEST THAT LOOPS
            ;2408 ; BACK TO THE BEGINNING, WITH MORE THAN ONE SUBTEST.
            ;2409 ;-
            ;2410 SUBTEST1:
U 11B9, 0810,0038,0180,0978,0000,11BA  ;2411  D_RC[0F]
U 11BA, 0019,0032,4D80,09F8,0000,0001  ;2412  ALU_D.OR.K[.FF00],RC[0F]_ALU,RETURN1
            ;2413 ;*
            ;2414 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2415 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2416 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2417 ;-
U 11BB, 0010,0038,01C0,0978,0000,11BC  ;2418 SETRCF: Q_RC[0F]
U 11BC, 0019,2034,4DC0,0800,0000,11BD  ;2419  Q_Q.AND.K[.FF00]
U 11BD, 0019,2032,1D80,09F8,0000,0001  ;2420  RC[0F]_Q.OR.SC,RETURN1
            ;2421 ;*****************************************************************
            ;2422 ;*
            ;2423 ;      THIS ROUTINE ENABLES SBI FAULT INTERRUPTS BY
            ;2424 ; FIRST CLEARING FAULT<19>, AND THEN SETTING FAULT<18>
            ;2425 ; IN THE SBI FAULT REGISTER (ID ADRS:1B)
            ;2426 ;
            ;2427 ; D AND SC GET CLOBBERED
            ;2428 ;-
            ;2429 ;*****************************************************************
U 11BE, 0F00,003C,2180,0800,0084,71BF  ;2430 ENBFLT: SC_K[.14],D_0
U 11BF, 0000,003C,0580,0800,0084,B1C0  ;2431  SC_SC-K[.1]
U 11C0, 0801,001C,0180,0800,0000,11C1  ;2432  D_D.ORNOT.MASK  ; FAULT<19> IS WRITE "1" TO CLEAR
U 11C1, 0600,003C,6D80,3C00,0000,11C2  ;2433  ID[FAULT]_D,D_D.RIGHT
U 11C2, 0000,003E,6D80,3C00,0000,0001  ;2434  ID[FAULT]_D,RETURN1
            ;2435
```

```
                ;2436  .PAGE
                ;2437  .TOC "    NOINTR ROUTINE"
                ;2438 ;*****************************************************************
                ;2439 ;*
                ;2440 ; THIS SUBROUTINE CHECKS FOR AN UNEXPECTED INTERRUPT.
                ;2441 ; FIRST, THE PSL IS CLEARED SO AS NOT TO MASK ANY LOWER
                ;2442 ; PRIORITY INTERRUPTS, THEN THE INTERRUPT IS STROBED.
                ;2443 ; IF ANY INTERRUPTS OCCURRED:
                ;2444 ;  RC[8] GETS THE VECTOR REGISTER (ID ADRS:0D)
                ;2445 ;  RC[7] GETS THE SBI ERROR REGISTER (ID ADRS:19)
                ;2446 ;  RC[6] GETS THE FAULT REGISTER(ID ADRS:1B)
                ;2447 ;  RC[5] GETS A 1 (I.E., INTERRUPT OCCURRED)
                ;2448 ;  RETURN1 IS USED TO GET BACK
                ;2449 ; ELSE
                ;2450 ;  RETURN2 IS USED
                ;2451 ;
                ;2452 ; THE D AND Q REGISTERS GET CLOBBERED.
                ;2453 ;
                ;2454 ;-
                ;2455 ;*****************************************************************
                ;2456 =0
U 1030. 0000.003D.0180.0800.0000.11C8   ;2457 NOINTR: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 1031. 0000.003C.35F0.2C00.0000.11C3   ;2458  Q_ID[VECTOR]
U 11C3. 0019.2034.8180.0800.0010.11C4   ;2459  ALU_Q.AND.K[.3FF],CLK.UBCC
U 11C4. 0000.013C.0180.0800.0000.1034   ;2460  Z?
U 1034. 0000.003C.0180.0800.0000.11CC   ;2461 =0 J/ERRFLT    ; INTR. OCCURRED,SET FLAG,GET DATA
U 1035. 0000.003E.0180.0800.0000.0002   ;2462  RETURN2     ; NO INTR, RETURN2
                ;2463 =0
U 1038. 0000.003D.0180.0800.0000.11C8   ;2464 WRTOUT: CALL[STROBE]    ; STROBE INTR.,WAIT DURING RETURN
U 1039. 0000.003C.35F0.2C00.0000.11C5   ;2465  Q_ID[VECTOR]
U 11C5. 0819.2034.8180.0800.0000.11C6   ;2466  ALU_Q.AND.K[.3FF],D_ALU
U 11C6. 0019.0020.A580.0800.0010.11C7   ;2467  ALU_D.XOR.K[.60],CLK.UBCC ; TIMEOUT VECTOR=60
U 11C7. 0000.013C.0180.0800.0000.103C   ;2468  Z?
U 103C. 0000.003C.0180.0800.0000.11CC   ;2469 =0 J/ERRFLT    ; WRONG VECTOR,SET FLAG,GET DATA
U 103D. 0000.003E.0180.0800.0000.0002   ;2470  RETURN2     ; OK, RETURN2
U 11C8. 0F00.003C.0180.0800.0000.11C9   ;2471 STROBE: D_0     ; SET LOWEST IPL ACTIVE
U 11C9. 0000.003C.3D80.3C00.0000.11CA   ;2472  ID[PSL]_D    ;
U 11CA. 0000.003C.0180.0800.4000.11CB   ;2473  IEK/ISTR    ; STROBE INTERRUPT
U 11CB. 0000.003E.0180.0800.0000.0001   ;2474  RETURN1     ; WAIT BEFORE LATCHING VECTOR REGISTER
U 11CC. 0018.0038.0580.09A8.0000.11CD   ;2475 ERRFLT: RC[5]_K[.1]    ; Set Flag
U 11CD. 0001.203C.65F0.2DC0.0000.11CE   ;2476  RC[8]_Q,Q_ID[SBI.ERR]  ; RC[8]_VECTOR
U 11CE. 0001.203C.6DF0.2DB8.0000.11CF   ;2477  RC[7]_Q,Q_ID[FAULT]   ; RC[7]_SBI.ERR
U 11CF. 0001.203E.0180.09B0.0000.0001   ;2478  RC[6]_Q,RETURN1    ; RC[6]_FAULT,RETURN TO TEST (WITH ERROR)
                ;2479
                ;2480 INVALIDATECACHE:
U 11D0. 0010.0038.0180.0970.0200.11D1   ;2481  VA_RC[0E]    ;SET ADDRESS
U 11D1. 0818.0038.4580.0800.0000.11D2   ;2482  D_K[.8000]    ;SET GROUP UNDER TEST
U 11D2. 0010.0038.0180.0968.0010.11D3   ;2483  ALU_RC[0D],CLK.UBCC
U 11D3. 0600.013C.0180.0800.0000.1040   ;2484  Z?,D_D.RIGHT    ;GROUP 0 UNDER TEST
U 1040. 0600.003C.0180.0800.0000.1041   ;2485 =0 D_D.RIGHT    ;GROUP 1 UNDER TEST
U 1041. 0000.003C.7580.3C00.0000.11D4   ;2486  ID[MAINT]_D    ;WRITE THE I.D. REGISTER
U 11D4. 0810.0038.0180.9170.0000.11D5   ;2487  MCT/INVALIDATE,D_RC[0E]  ;WRITE CACHE
U 11D5. 0000.003C.8580.0800.0084.71D6   ;2488  SC_K[.C]
U 11D6. 0001.0024.0180.0800.0010.11D7   ;2489  ALU_D.ANDNOT.MASK,CLK.UBCC ;BIT #12 SET IN ADDRESS ??
U 11D7. 0000.013C.0180.0800.0000.1044   ;2490  Z?
```

```
U 1044, 0000,003E,0180,0800,0000,0001  ;2491 =0 RETURN1     ; AT MAX
U 1045, 0819,0014,1180,09F0,0000,11D0  ;2492  RC[0E]_ALU,D_D+K[.4],J/INVALIDATECACHE ;INCREMENT AND REPEAT
            ;2493
U 11D8, 0818,0038,0180,0800,0000,105E  ;2494 DELAY: D_K[.8],J/CALLCON
            ;2495
            ;2496
            ;2497
```

```
            ;2498  .PAGE
            ;2499  .TOC "    Wait Loop Routine"
            ;2500 ;++
            ;2501 ; FUNCTIONAL DESCRIPTION:
            ;2502 ;
            ;2503 ; This routine is used to wait the specified number of machine cycles.
            ;2504 ; This routine is typically called to wait for an SBI write to
            ;2505 ; I/O space to complete.
            ;2506 ;
            ;2507 ; CALLING CONSTRAINT:
            ;2508 ;
            ;2509 ; =0
            ;2510 ;
            ;2511 ; INPUTS:
            ;2512 ;
            ;2513 ; d - Contains number of cycles to wait
            ;2514 ; alu.z condition code must not be set
            ;2515 ;
            ;2516 ; SIDE EFFECTS:
            ;2517 ;
            ;2518 ; d is destroyed
            ;2519 ;--
            ;2520 WAIT.LOOP:
U 11D9, 0018,0038,6180,0800,0010,1048  ;2521  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2522      ; ...is not set
            ;2523 =0
            ;2524 W.LOOP:
            ;2525  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 1048, 0819,0100,0580,0800,0010,1048  ;2526       j/W.LOOP
U 1049, 0000,003E,0180,0800,0000,0001  ;2527  return1    ; exit
            ;2528
            ;2529 SBI.UNJAM:
            ;2530 CLRSBI:
            ;2531 ;=========================================================================
            ;2532 ;/////////////////////////////////////////////////////////////////////////
            ;2533 ;+
            ;2534 ; THIS SUBROUTINE IS USED TO UNJAM THE SBI.
            ;2535 ; HOLD IS ASSERTED FOR 16 CYCLES.
            ;2536 ; THEN HOLD AND UNJAM ARE ASSERTED FOR 16 CYCLES.
            ;2537 ; FINALLY HOLD ALONE IS ASSERTED FOR 16 CYCLES.
            ;2538 ;-
U 11DA, 0818,0038,6580,8000,0010,104C  ;2539 UNJAM: MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2540 =0
U 104C, 0819,0100,0580,8000,0010,104C  ;2541 UNJAMA: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMA
U 104D, 0818,0038,6580,8800,0010,1058  ;2542  MCT/SBI.HOLD+UNJAM,D_K[.10],CLK.UBCC ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
            ;2543 =0
U 1058, 0819,0100,0580,8800,0010,1058  ;2544 UNJAMB: MCT/SBI.HOLD+UNJAM,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMB
U 1059, 0818,0038,6580,8000,0010,105C  ;2545  MCT/SBI.HOLD,D_K[.10],CLK.UBCC ; ASSERT HOLD FOR 16 CYCLES.
            ;2546 =0
U 105C, 0819,0100,0580,8000,0010,105C  ;2547 UNJAMC: MCT/SBI.HOLD,D_D-K[.1],CLK.UBCC,Z?,J/UNJAMC
U 105D, 0000,003E,0180,0800,0000,0001  ;2548  RETURN1    ; DONE.
            ;2549 INIT: ;============================================================
            ;2550  ;
            ;2551  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2552  ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
```

```
            ;2553  ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2554  ;
            ;2555  ;=====================================================================
            ;2556  ;
U 11DB, 0003,003C,0180,0AF8,0000,10A0  ;2557  R[0F]_0     ;CLEAR U-TRAPS
U 10A0, 0000,003D,0180,0800,0000,11DA  ;2558 =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 10A1, 0000,003C,0180,0800,0000,10A2  ;2559  NOP
U 10A2, 0000,003D,0180,0800,0000,11DE  ;2560 =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2561        ;  INTRUPT REG
U 10A3, 0000,003C,0180,0800,0000,10A4  ;2562  NOP
U 10A4, 0000,003D,0180,0800,0000,11E5  ;2563 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 10A5, 0000,003C,0180,0800,0000,10A6  ;2564  NOP
U 10A6, 0000,003D,0180,0800,0000,11E9  ;2565 =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 10A7, 0000,003C,0180,0800,0000,10A8  ;2566  NOP
U 10A8, 0000,003D,0180,0800,0000,11E1  ;2567 =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 10A9, 0000,003C,0180,0800,0000,10AA  ;2568  NOP
U 10AA, 0000,003D,0180,0800,0000,11F1  ;2569 =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 10AB, 0000,003C,0180,0800,0000,10AC  ;2570  NOP
U 10AC, 0000,003D,0180,0800,0000,11BE  ;2571 =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 10AD, 0818,0038,4580,0800,0000,10AE  ;2572  D_K[.8000]    ;ENABLE CACHE PARITY ERROR
U 10AE, 0000,003D,7980,3C00,0000,11ED  ;2573 =0 CALL[CACOFF],ID[PARITY]_D ;SHUT CACHE OFF
U 10AF, 0F00,003C,0180,0800,0000,11DC  ;2574  D_0
U 11DC, 0000,003C,4980,3C00,0000,11DD  ;2575  ID[TBER0]_D    ;SHUT TB OFF
U 11DD, 0000,003E,7180,3C00,0000,0001  ;2576  ID[COMP]_D,RETURN1  ;DISABLE SILO INTERRUPTS
            ;2577 SETPSL: ;=====================================================================
            ;2578  ;
            ;2579  ;DROP THE CPU IPR LEVEL TO
            ;2580  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2581  ; PENDING
            ;2582  ;
            ;2583  ;=====================================================================
            ;2584  ;
U 11DE, 0F00,003C,0180,0800,0000,11DF  ;2585  D_0
U 11DF, 0000,003C,3980,3C00,0000,11E0  ;2586  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 11E0, 0000,003E,3D80,3C00,0000,0001  ;2587  RETURN1,ID[PSL]_D
            ;2588 CLRTIM: ;=====================================================================
            ;2589  ;
            ;2590  ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2591  ; SBI ERROR REGISTER
            ;2592  ;
            ;2593  ;=====================================================================
            ;2594  ;
U 11E1, 0818,0038,0580,0800,0000,11E2  ;2595  D_K[.1]
U 11E2, 0000,003C,85F8,0800,0084,71E3  ;2596  Q_0,SC_K[.C]
U 11E3, 0D00,003C,0180,0800,0000,11E4  ;2597  D_DAL.SC
U 11E4, 0000,003E,6580,3C00,0000,0001  ;2598  ID[SBI.ERR]_D,RETURN1
            ;2599 CLRFLT: ;=====================================================================
            ;2600  ;
            ;2601  ;CLEAR THE CP FAULT BIT IN THE
            ;2602  ; FAULT REGISTER
            ;2603  ;
            ;2604  ;=====================================================================
            ;2605  ;
U 11E5, 0818,0038,0180,0800,0000,11E6  ;2606  D_K[.8]
U 11E6, 0000,003C,65F8,0800,0084,71E7  ;2607  Q_0,SC_K[.10]
```

```
U 11E7, 0D00,003C,0180,0800,0000,11E8  ;2608  D_DAL.SC
U 11E8, 0000,003E,6D80,3C00,0000,0001  ;2609  ID[FAULT]_D,RETURN1
            ;2610 CLRECC: ;=================================================================
            ;2611  ;
            ;2612  ;CLEAR CRD,RDS BIT IN THE
            ;2613  ;SBI ERROR REGISTER
            ;2614  ;
            ;2615  ;=================================================================
            ;2616  ;
U 11E9, 0818,0038,0D80,0800,0000,11EA  ;2617  D_K[.3]
U 11EA, 0000,003C,89F8,0800,0084,71EB  ;2618  Q_0,SC_K[.D]
U 11EB, 0D00,003C,0180,0800,0000,11EC  ;2619  D_DAL.SC
U 11EC, 0000,003E,6580,3C00,0000,0001  ;2620  ID[SBI.ERR]_D,RETURN1
            ;2621 DISABLE.CACHE:
            ;2622 CACOFF: ;=================================================================
            ;2623  ;
            ;2624  ;FORCES MISSES IN THE CACHE
            ;2625  ;   ^X18000 TO THE MAINT REGISTER
            ;2626  ;
            ;2627  ;=================================================================
            ;2628  ;
U 11ED, 0818,0038,0D80,0800,0000,11EE  ;2629  D_K[.3]
U 11EE, 0000,003C,61F8,0800,0084,71EF  ;2630  Q_0,SC_K[.F]   ;15 SHIFTS TO THE LEFT
U 11EF, 0D00,003C,0180,0800,0000,11F0  ;2631  D_DAL.SC
U 11F0, 0000,003E,7580,3C00,0000,0001  ;2632  ID[MAINT]_D,RETURN1
            ;2633 ENBECC: ;=================================================================
            ;2634  ;
            ;2635  ;ENABLE CRD,RDS INTERRUPTS
            ;2636  ;
            ;2637  ;=================================================================
            ;2638  ;
U 11F1, 0818,0038,0580,0800,0000,11F2  ;2639  D_K[.1]
U 11F2, 0000,003C,61F8,0800,0084,71F3  ;2640  Q_0,SC_K[.F]
U 11F3, 0D00,003C,0180,0800,0000,11F4  ;2641  D_DAL.SC
U 11F4, 0000,003E,6580,3C00,0000,0001  ;2642  ID[SBI.ERR]_D,RETURN1
            ;2643
```

```
                ;2644  .PAGE
                ;2645  .TOC "   FIND MEMORY AT TR"
                ;2646 ;*****************************************************
                ;2647 ;++
                ;2648 ;
                ;2649 ; Functional Description:
                ;2650 ; ----------------------
                ;2651 ;
                ;2652 ; This subroutine looks at the present TR level for a
                ;2653 ; memory system (either MS780-A/C or MS780-E/H).
                ;2654 ;
                ;2655 ;
                ;2656 ;
                ;2657 ; Calling Constraint: =00
                ;2658 ; ------------------
                ;2659 ;
                ;2660 ;
                ;2661 ; Inputs:
                ;2662 ; ------
                ;2663 ;
                ;2664 ; ID REGISTER 24 MUST HOLD THE TR LEVEL
                ;2665 ;
                ;2666 ;
                ;2667 ; Outputs:
                ;2668 ; -------
                ;2669 ;
                ;2670 ; RETURN1 IF NO MEMORY AT THIS TR
                ;2671 ; R[0A] AND R[0B] ARE SET UP  AS DESCRIBED IN THE
                ;2672 ;  HEADER FOR THE ROM TEST
                ;2673 ;
                ;2674 ; Side Effects:
                ;2675 ; ------------
                ;2676 ;
                ;2677 ; The contents of the following registers are lost:
                ;2678 ;
                ;2679 ; D, Q, SC, VA, R[0A], R[0B]
                ;2680 ;
                ;2681 ;--
                ;2682 ;*****************************************************
                ;2683 FNDMEM:
U 11F5, 0000,003C,91F0,2CC0,0000,10B0  ;2684  Q_ID[24]   ; Get TR level
                ;2685 =0 D_Q,     ; Generate IO Addres
U 10B0, 0C00,003D,0180,0800,0000,128C  ;2686   CALL[GENERATE.IO] ; ...
                ;2687  VA_ALU,ALU_D,    ; Get IO Address to the VA
U 10B1, 0001,003C,8980,0800,0284,71F6  ;2688   SC_K[.D]  ; Prepare to enable Time-Out uTraps
U 11F6, 0003,001C,0180,0AF8,0000,11F7  ;2689  R[0F]_NOT.MASK   ; Enable Time-Out uTraps
U 11F7, 0000,003C,0180,C800,0000,11F8  ;2690  D[LONG]_CACHE.P   ; Read CNFG Reg
                ;2691  ALU_R[0F],    ; Was there a response from this TR
U 11F8, 0000,003C,0180,0A78,0010,11F9  ;2692   CLK.UBCC  ; ...level?
U 11F9, 0000,013C,0180,0800,0000,10B2  ;2693  Z?
U 10B2, 0000,003C,0180,0800,0000,11FA  ;2694 =0 J/FOUND.SOME.DEVICE  ; Found some device at this TR
U 10B3, 0000,003E,0180,0800,0000,0001  ;2695  RETURN1    ; No Device at this TR
                ;2696 FOUND.SOME.DEVICE:
                ;2697  RC[0]_ALU.RIGHT,SI/ZERO, ; Save received data from CNFG Reg
U 11FA, 0041,003C,0180,0980,0000,11FB  ;2698   ALU_D   ; ..but shifted right one bit
```

```
U 11FB. 0001.003C.0180.0800.0082.11FC  ;2699  SC_D          ; Get ready to branch on the adapter code
U 11FC. 0000.003C.0180.0800.0100.11FD  ;2700  FE_SC      ; ..Load FE
U 11FD. 0000.003C.0180.0800.1400.71FE  ;2701  STATE_FE    ; ..Load the STATE
U 11FE. 0000.163C.0180.0800.0000.1019  ;2702  STATE7-4?    ; What kind of device did we find?
          ;2703 =1001 R[0B]_K[.2],    ; Found a MS780-A/C
U 1019. 0018.0038.0980.0AD8.0000.1204  ;2704   J/FNDMEM.MS780AC ; ...
U 101B. 0000.003E.0180.0800.0000.0001  ;2705  RETURN1     ; Not a memory
U 101D. 0000.003E.0180.0800.0000.0001  ;2706  RETURN1     ; Not a memory
          ;2707  R[0B]_K[.1],    ; Found a MS780-E/H
U 101F. 0018.0038.0580.0AD8.0000.11FF  ;2708   J/FNDMEM.MS780EH ; ...
          ;2709 FNDMEM.MS780EH:
U 11FF. 0810.0038.0180.0900.0000.1200  ;2710  D_RC[0]     ; Get the contents of CNFG Reg A
U 1200. 0001.003C.0180.0800.0082.1201  ;2711  SC_D     ; Get ready to branch on the chip type
U 1201. 0000.003C.0180.0800.0100.1202  ;2712  FE_SC      ; Load the FE
U 1202. 0000.003C.0180.0800.1400.7203  ;2713  STATE_FE    ; Load the STATE
U 1203. 0000.173C.0180.0800.0000.1023  ;2714  STATE3-0?    ; Branch on the chip type
U 1023. 0000.003D.0180.0800.0000.119D  ;2715 =0011 ERROR1     ; Mixed arrays (64K and 256K)
          ;2716  R[0A]_K[.2],    ; Found 64K chips.
U 1027. 0018.003A.0980.0AD0.0000.0002  ;2717   RETURN2    ; ...return
          ;2718  R[0A]_K[.3],    ; Found 256K chips
U 102B. 0018.003A.0D80.0AD0.0000.0002  ;2719   RETURN2    ; ...return to caller
U 102F. 0000.003D.0580.0800.0084.7024  ;2720  ERROR1[.1]    ; Missing arrays
          ;2721 FNDMEM.MS780AC:
U 1204. 0810.0038.0180.0900.0000.1205  ;2722  D_RC[0]     ; Get contents of CNFG Reg A
U 1205. 0001.003C.0180.0800.0082.1206  ;2723  SC_D     ; Prepare to branch on the chip type
U 1206. 0000.003C.0180.0800.0100.1207  ;2724  FE_SC      ; Load the FE
U 1207. 0000.003C.0180.0800.1400.7208  ;2725  STATE_FE    ; Load the STATE
U 1208. 0000.173C.0180.0800.0000.1033  ;2726  STATE3-0?
U 1033. 0000.003D.0580.0800.0084.7024  ;2727 =0011 ERROR1[.1]    ; Missing arrays
          ;2728  R[0A]_K[ZERO],    ; Found 4K chips
U 1037. 0018.003A.1980.0AD0.0000.0002  ;2729   RETURN2    ; ...return to caller
          ;2730  R[0A]_K[.1],    ; Found 16K chips
U 103B. 0018.003A.0580.0AD0.0000.0002  ;2731   RETURN2    ; ...return to caller
U 103F. 0000.003D.0580.0800.0084.7024  ;2732  ERROR1[.1]    ; Mixed arrays (4K and 16 K chips)
          ;2733
          ;2734
          ;2735 ;////////////////////////////////////////////////////////////////
          ;2736 ;*
          ;2737 ;       THIS SUBROUTINE IS USED TO SEE IF THE ROMS ON AN MS780
          ;2738 ; ARE ENABLED.
          ;2739 ;  ENABLED=>RETURN2
          ;2740 ;  DISABLED=>RETURN1
          ;2741 ;-
U 1209. 0000.003C.91F0.2C00.0000.120A  ;2742 FNDROM: Q_ID[24]
U 120A. 0C00.003C.0180.0800.0000.10B4  ;2743  D_Q
U 10B4. 0000.003D.0180.0800.0000.128C  ;2744 =0 CALL[GENERATE.IO]  ; Generate IO Base address of this
          ;2745       ; ...controller
U 10B5. 0000.003C.8580.0800.0084.720B  ;2746  SC_K[.C]    ;
          ;2747  VA_ALU,ALU_D.ORNOT.MASK,SC_SC+1,; SC=D
U 120B. 0001.001C.0180.09D8.0280.D20C  ;2748  RC[0B]_ALU    ; Save Base Address of the roms
U 120C. 0003.001C.0180.CAF8.0000.120D  ;2749  MCT/READ.P,R[0F]_NOT.MASK
U 120D. 0000.003C.0180.0A78.0010.120E  ;2750  ALU_R[0F],CLK.UBCC
U 120E. 0000.013C.0180.0800.0000.10B6  ;2751  Z?     ; ROM RESPONSE?
U 10B6. 0000.003E.0180.0800.0000.0002  ;2752 =0 RETURN2     ; YES, NO T.O.
U 10B7. 0000.003E.0180.0800.0000.0001  ;2753  RETURN1     ; NO, T.O. OCCURRED
```

```
            ;2754
            ;2755
            ;2756 ;////////////////////////////////////////////////////////////////
            ;2757 ;*
            ;2758 ;       THIS ROUTINE IS USED TO ADD UP ALL OF THE ROM LOCATIONS
            ;2759 ; ON THE MS780, FINDING THE CHECKSUM.
            ;2760 ; THE CHECKSUM IS RETURNED IN THE Q-REGISTER
            ;2761 ;-
U 120F, 0810,0038,0180,0958,0000,1210  ;2762 ADDUP: D_RC[0B]    ; Get ROM Starting address
U 1210, 0001,003C,0180,0800,0200,1211  ;2763  VA_ALU,ALU_D     ; Get it to the VA
U 1211, 0000,003C,8980,0800,0084,7212  ;2764  SC_K[.D]
U 1212, 0003,003C,01F8,0A88,0000,1213  ;2765  R[1]_0,Q_0       ; R[1]=DONE FLAG,Q=SUM
            ;2766
U 1213, 0003,001C,0180,CAF8,0000,1214  ;2767 ADLP: MCT/READ.P,R[0F]_NOT.MASK ; SC=D FOR TRAP CATCHER
U 1214, 001D,0014,05C0,0A08,0084,B215  ;2768  Q_ALU,ALU_D+Q,LAB_R[1],SC_SC-K[.1] ; SC=C FOR DONE FLAG
U 1215, 0018,0014,1180,0A8B,0000,1216  ;2769  R[1]_ALU,ALU_LA+K[.4],VA_VA+4
U 1216, 0000,0024,0180,0A08,0010,1217  ;2770  ALU_R[1].ANDNOT.MASK,CLK.UBCC ; DONE YET?
U 1217, 0000,013C,0180,0800,0000,10B8  ;2771  Z?
U 10B8, 0000,003E,0180,0800,0000,0001  ;2772 =0 RETURN1
U 10B9, 0000,003C,0180,0800,0080,D213  ;2773  J/ADLP,SC_SC+1
            ;2774
            ;2775 ;////////////////////////////////////////////////////////////////
            ;2776 ;*
            ;2777 ;       THIS ROUTINE IS USED TO COMPUTE THE EXPECTED DATA
            ;2778 ; CONTAINED IN THE FIRST ROM LOCATION ON THE MS780.
            ;2779 ;
            ;2780 ; THIS DATA WILL PROBABLY CHANGE AS NEW DEVICES ARE
            ;2781 ; INTRODUCED.
            ;2782 ;
            ;2783 ; THE EXPECTED DATA AS OF 12/1/78 IS 071159D4
            ;2784 ;-
U 1218, 001B,0000,0180,0800,0082,1219  ;2785 FRSLOC: SC_ALU,ALU_0-K[.8] ;
U 1219, 0018,0038,29C0,0800,0000,121A  ;2786  Q_K[.34]
U 121A, 0000,003C,2580,0800,0000,121B  ;2787  K[.A0]
U 121B, 0019,2014,25C0,0800,0000,121C  ;2788  Q_Q+K[.A0]
U 121C, 0D00,003C,0180,0800,0000,121D  ;2789  D_DAL.SC   ; D4000000
U 121D, 0018,0038,A5C0,0800,0000,121E  ;2790  Q_K[.60]
U 121E, 0000,003C,5D80,0800,0000,121F  ;2791  K[.7]
U 121F, 0019,2000,5DC0,0800,0000,1220  ;2792  Q_Q-K[.7]
U 1220, 0D00,003C,6580,0800,0000,1221  ;2793  D_DAL.SC,K[.10]  ; 59D40000
U 1221, 001B,0010,65C0,0800,0000,1222  ;2794  Q_ALU,ALU_0+K[.10]+1
U 1222, 0D00,003C,0180,0800,0000,1223  ;2795  D_DAL.SC   ; 1159D400
U 1223, 0018,0038,5DC0,0800,0000,1224  ;2796  Q_K[.7]
U 1224, 0D00,003E,0180,0800,0000,0001  ;2797  D_DAL.SC,RETURN1 ; 071159D4
            ;2798
            ;2799 ;////////////////////////////////////////////////////////////////
            ;2800 ;*
            ;2801 ;       THIS ROUTINE GENERATES THE EXPECTED ROM 32-BIT
            ;2802 ; CHECKSUM.
            ;2803 ;
            ;2804 ; THIS DATA WILL DEFINITELY CHANGE AS NEW DEVICES ARE INTRODUCED
            ;2805 ; ON THE SBI.
            ;2806 ;
            ;2807 ; THE EXPECTED 32-BIT CHECKSUM AS OF 12/1/78 =25CCD7E5
            ;2808 ;-
```

```
            ;2809 MSC.CHECKSUM:
U 1225. 001B.0000.0180.0800.0082.1226  ;2810  SC_ALU,ALU_0-K[.8]
U 1226. 0018.0038.3DC0.0800.0000.1227  ;2811  Q_K[.EF]
U 1227. 0019.2020.F5C0.0800.0000.1228  ;2812  Q_ALU,ALU_Q.XOR.K[.A]
U 1228. 0D00.003C.0180.0800.0000.1229  ;2813  D_DAL.SC  ; E5000000
U 1229. 0018.0038.D1C0.0800.0000.122A  ;2814  Q_K[.C0]
U 122A. 0019.2030.65C0.0800.0000.122B  ;2815  Q_Q.OR.K[.10]
U 122B. 0019.2030.5DC0.0800.0000.122C  ;2816  Q_Q.OR.K[.7]
U 122C. 0D00.003C.0180.0800.0000.122D  ;2817  D_DAL.SC  ; D7E50000
U 122D. 0018.0038.D1C0.0800.0000.122E  ;2818  Q_K[.C0]
U 122E. 0019.2030.85C0.0800.0000.122F  ;2819  Q_Q.OR.K[.C]
U 122F. 0D00.003C.0180.0800.0000.1230  ;2820  D_DAL.SC  ; CCD7E500
U 1230. 0000.003C.E980.0800.0000.1231  ;2821  K[.24]
U 1231. 001B.0010.E9C0.0800.0000.1232  ;2822  Q_ALU,ALU_0+K[.24]+1
U 1232. 0D00.003E.0180.0800.0000.0001  ;2823  D_DAL.SC,RETURN1 ; 25CCD7E5
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 1871
ESKARS3.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:55                    Page   7

            ;2824  .page
            ;2825
            ;2826
            ;2827 ;////////////////////////////////////////////////////////////////////////
            ;2828 ;*
            ;2829 ;       THIS ROUTINE READS A ROM LOCATION, THAT (AS OF 12-5-78)
            ;2830 ; CONTAINS ALL ZEROES.
            ;2831 ; THE ADDRESS = BASE ADDRESS OF TR * 1800
            ;2832 ; FOR EXAMPLE, THE TR1 ADDRESS WOULD BE 20003800.
            ;2833 ; THIS MAY CHANGE AS NEW DEVICES ARE INTRODUCED
            ;2834 ; ONTO THE SBI.
            ;2835 ; THE ROUTINE WILL RETURN1 IF:
            ;2836 ;  THAT LOCATION DOES NOT = 0
            ;2837 ; AND WILL RETURN2 IF:
            ;2838 ;  THAT LOCATION = 0.
            ;2839 ;-
U 1233, 0810.0038.0180.0970.0000.10BA  ;2840 RDZERO: D_RC[0E]
U 10BA, 0000.003D.0180.0800.0000.128C  ;2841 =0 CALL[GENERATE.IO]  ; Generate IO address based on TR
U 10BB, 0001.003C.F580.0980.0000.1234  ;2842  RC[0]_D,K[.A]   ; Save IO Base address
U 1234, 001B.0010.F580.0800.0082.1235  ;2843  SC_ALU,ALU_0+K[.A]+1  ; Form address 2000x800 in D reg
U 1235, 0818.0038.0DF8.0800.0000.1236  ;2844  D_K[.3],Q_0   ; ...
U 1236, 0D00.003C.0180.0800.0000.1237  ;2845  D_DAL.SC    ; ...
U 1237, 0011.0020.8980.0900.0284.7238  ;2846  VA_ALU,ALU_D.XOR.RC[0],SC_K[.D] ; Put address in VA
            ;2847  D[LONG]_CACHE.P,  ; Read location
U 1238, 0003.001C.0180.CAF8.0000.1239  ;2848   R[0F]_NOT.MASK  ; Enable Time-Out traps
U 1239, 0001.003C.0180.09E0.0010.123A  ;2849  RC[0C]_D,CLK.UBCC  ; Save returned data an return
U 123A, 0000.013C.0180.0800.0000.10BC  ;2850  Z?    ; ...depending on whether it was
            ;2851       ; ...zero or not
U 10BC, 0000.003E.0180.0800.0000.0001  ;2852 =0 RETURN1    ; DATA RETURNED WAS NOT ZERO
U 10BD, 0000.003E.0180.0800.0000.0002  ;2853  RETURN2    ; OK.  DATA RECEIVED IS ZERO
            ;2854
            ;2855 ;////////////////////////////////////////////////////////////////////////
            ;2856 ;*
            ;2857 ;       THIS ROUTINE READS TWO ROM LOCATIONS, THAT, TOGETHER
            ;2858 ; READ A "1" FROM EVERY BIT.
            ;2859 ; THESE TWO LOCATIONS (AS OF 12-5-78) ARE:
            ;2860 ; TR BASE ADDRESS +1010 (E.G TR1 = 20003010)
            ;2861 ;  CONTAINS 51FFFFFF
            ;2862 ; AND
            ;2863 ; TR BASE ADDRESS +131C (E.G. TR1 = 2000331C)
            ;2864 ;  CONTAINS EFDE5212
            ;2865 ;
            ;2866 ; THIS DATA MAY CHANGE AS NEW DEVICES ARE INTRODUCED ONTO THE
            ;2867 ; SBI.
            ;2868 ;
            ;2869 ; THE ROUTINE WILL RETURN1 IF :
            ;2870 ;  THE CONTENTS OF THESE TWO LOCATIONS DOES NOT "OR" TO -1
            ;2871 ; AND WILL RETURN2 IF:
            ;2872 ;  THE CONTENTS OF THESE TWO LOCATIONS DOES "OR" TO -1
            ;2873 ;-
            ;2874 MSC.ALLONE:
U 123B, 0000.003C.0180.0800.0084.723C  ;2875  SC_K[.8]
U 123C, 0818.0038.65C0.0800.0000.123D  ;2876  D_ALU,Q_ALU,ALU_K[.10]
U 123D, 0D00.003C.0180.0800.0000.123E  ;2877  D_DAL.SC
U 123E, 081D.0030.0180.0800.0000.123F  ;2878  D_D.OR.Q  ; D=1010
```

```
U 123F. 0001.003C.91F0.2D80.0000.1240  ;2879  RC[0]_D,Q_ID[24] ; SAVE THAT OFFSET, GET TR LEVEL
U 1240. 0C00.003C.0180.0800.0000.10BE  ;2880  D_Q
U 10BE. 0000.003D.0180.0800.0000.128C  ;2881 =0 CALL[GENERATE.IO] ; GET TR BASE ADDRESS INTO D
U 10BF. 0011.0030.89C0.0900.0284.7241  ;2882  VA_ALU,Q_D.OR.RC[0],SC_K[.D] ; VA_ADRS,Q_SAVED ADRS
U 1241. 0003.001C.0180.CAF8.0000.1242  ;2883  MCT/READ.P,R[0F]_NOT.MASK
U 1242. 0001.003C.0180.0980.0000.1243  ;2884  RC[0]_D   ; SAVE THAT DATA
          ;2885
U 1243. 0878.0038.D180.0800.0000.1244  ;2886  D_ALU.LEFT2,ALU_K[.C0],SI/ZERO ; OR IN THE NEXT ADRS
U 1244. 0819.0030.8580.0800.0000.1245  ;2887  D_D.OR.K[.C]
U 1245. 001D.0030.9980.0800.0284.7246  ;2888  VA_ALU,ALU_D.OR.Q,SC_K[.D]
U 1246. 0003.001C.0180.CAF8.0000.1247  ;2889  MCT/READ.P,R[0F]_NOT.MASK
U 1 .7. 0811.0030.0180.0900.0000.1248  ;2890  D_D.OR.RC[0]  ; OR THE TWO LOCATIONS' DATA TOGETHER
U 1248. 0001.003C.0180.09E0.0000.1249  ;2891  RC[0C]_D  ; SAVE GOT DATA
U 1249. 001F.2010.0180.0800.0010.124A  ;2892  ALU_0+D+1,CLK.UBCC ; SEE IF IT ORED TO ALL 1'S
U 124A. 0000.013C.0180.0800.0000.10C0  ;2893  Z?
U 10C0. 0000.003E.0180.0800.0000.0001  ;2894 =0 RETURN1   ; NO, RETURN TO ERROR CALL
U 10C1. 0000.003E.0180.0800.0000.0002  ;2895  RETURN2   ; YES, OK.
          ;2896
```

```
;2897  .PAGE
;2898  .TOC "   OR FOR ONES ON MS780E"
;2899 ;*************************************************************
;2900 ;**
;2901 ;
;2902 ; Functional Description:
;2903 ; ----------------------
;2904 ;
;2905 ; This subroutine reads contents of locations 10AC and
;2906 ; 10B0 (relative to the I/O base address of the MS780E
;2907 ; memory under  test)  and ORs the contents of the two
;2908 ; locations.  If the  result is equal to all ones then
;2909 ; a RETURN2 is made, otherwise  a  RETURN1 is  made to
;2910 ; the calling routine.
;2911 ;
;2912 ;
;2913 ; Calling Constraint: =00
;2914 ; ------------------
;2915 ;
;2916 ;
;2917 ; Inputs:
;2918 ; ------
;2919 ;
;2920 ; ID register 24 must hold the TR level of the memory under test
;2921 ;
;2922 ;
;2923 ; Outputs:
;2924 ; -------
;2925 ;
;2926 ; RETURN1 if the result of the OR operation does NOT return all ones
;2927 ; RETURN2 if the result of the OR operation returns all ones
;2928 ; The result of the OR operation is saved in RC[0C]
;2929 ;
;2930 ; Side Effects:
;2931 ; ------------
;2932 ;
;2933 ; The contents of the following registers will be lost:
;2934 ;
;2935 ; D, Q, SC, RC[0], RC[1], VA
;2936 ;
;2937 ;--
;2938 ;*************************************************************
;2939 MSE.ALLONE:
U 124B, 0000,003C,91F0,2C00,0000,10C2  ;2940  Q_ID[24]   ; Get Present TR level
        ;2941 =0 D_Q,     ; Generate the IO Base address
U 10C2, 0C00,003D,0180,0800,0000,128C  ;2942   CALL[GENERATE.IO] ; ...
        ;2943  Q_D,     ; Save in the Q reg.
U 10C3, 0000,003C,85E0,0800,0084,724C  ;2944   SC_K[.C]  ; Prepare to set bit 12
U 124C, 0801,201C,0180,0800,0000,124D  ;2945  D_Q.ORNOT.MASK   ; Point to the first location
        ;2946       ; ...of ROM on this Controller
U 124D, 0001,003C,0180,0980,0000,124E  ;2947  RC[0]_D    ; Save in RC[0]
U 124E, 0819,0030,9580,0800,0000,124F  ;2948  D_D.OR.K[.B0]   ; Point to location 2000x0B0
        ;2949  VA_ALU,ALU_D,   ; Get address to the VA
U 124F, 0001,003C,8980,0800,0284,7250  ;2950   SC_K[.D]  ; Enable time-Out traps
        ;2951  D[LONG]_CACHE.P,  ; Read location 2000x0B0
```

```
U 1250, 0003,001C,0180,CAF8,0000,1251  ;2952   R[0F]_NOT.MASK  ; Enable Time-Out traps
U 1251, 0001,003C,0180,0988,0000,1252  ;2953  RC[1]_D     ; Save data read in RC[1]
U 1252, 0818,0038,9580,0800,0000,1253  ;2954  D_K[.B0]    ; Prepare to form 2000x0AC in
          ;2955       ; ...the D reg
U 1253, 0819,0000,6580,0800,0000,1254  ;2956  D_D-K[.10]    ; ...D =  A0
U 1254, 0819,0030,8580,0800,0000,1255  ;2957  D_D.OR.K[.C]    ; ...D = AC
U 1255, 0811,0030,0180,0900,0000,1256  ;2958  D_D.OR.RC[0]    ; ...D = 2000x0AC
          ;2959  VA_ALU,ALU_D,    ; Get address to the VA
U 1256, 0001,003C,8980,0800,0284,7257  ;2960   SC_K[.D]  ; Prepare to enable Tim-Out uTraps
          ;2961  D[LONG]_CACHE.P,   ; Read the location
U 1257, 0003,001C,0180,CAF8,0000,1258  ;2962   R[0F]_NOT.MASK  ; Enable Time-Out uTraps
U 1258, 0811,0030,0180,0908,0000,1259  ;2963  D_D.OR.RC[1]    ; OR the longwords read from ROM
          ;2964  ALU_0+D+1,    ; Is the result equal to xFFFFFFFF?
U 1259, 001F,2010,0180,0800,0010,125A  ;2965   CLK.UBCC  ; ...
U 125A, 0001,013C,0180,09E0,0000,10C4  ;2966  Z?,RC[0C]_D    ; Save result of OR for error logout
U 10C4, 0000,003E,0180,0800,0000,0001  ;2967 =0 RETURN1     ; Wrong data read form ROM
U 10C5, 0000,003E,0180,0800,0000,0002  ;2968  RETURN2     ; OK.
          ;2969
```

```
            ;2970  .PAGE
            ;2971  .TOC "   RESET SUBTEST NUMBER"
            ;2972 ;**
            ;2973 ; FUNCTIONAL DESCRIPTION:
            ;2974 ;
            ;2975 ; Since some  of the tests will have to be restarted when two
            ;2976 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2977 ; the subtest counter to zero ( only for thoes tests that have
            ;2978 ; more than one subtest). This subroutine may also be necessary
            ;2979 ; when a test is being loop on.
            ;2980 ;
            ;2981 ; CALLING CONSTRAINT:
            ;2982 ;
            ;2983 ; =0
            ;2984 ; SIDE EFFECTS:
            ;2985 ;
            ;2986 ; D is destroyed
            ;2987 ;
            ;2988 ;--
            ;2989 RESET.SUBTST:
            ;2990  RC[0F] 0,   ; Reset subtest number
U 125B, 0003,003E,0180,09F8,0000,0001  ;2991   RETURN1   ; ...and return
            ;2992
```

```
            ;2993  .PAGE
            ;2994  .TOC "   Initialize the SBI"
            ;2995 ;++
            ;2996 ; FUNCTIONAL DESCRIPTION:
            ;2997 ;
            ;2998 ; This routine performs the following functions:
            ;2999 ;
            ;3000 ;  Executes an SBI Unjam
            ;3001 ;  Clears the SBI Fault Latch
            ;3002 ;  Clears the SBI Error Register
            ;3003 ;
            ;3004 ; CALLING CONSTRAINT:
            ;3005 ;
            ;3006 ; =0
            ;3007 ;
            ;3008 ; SIDE EFFECTS:
            ;3009 ;
            ;3010 ; D & Q Register destroyed
            ;3011 ;--
            ;3012 =0
            ;3013 SBI.INIT:
U 10C6. 0000.003D.0180.0800.0000.11DA   ;3014  call[sbi.unjam]  ; Unjam the SBI
U 10C7. 0818.0038.C180.0800.0000.125C   ;3015  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 125C. 0801.0028.6580.3C00.0000.125D   ;3016  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 125D. 0F00.003C.6D80.3C00.0000.125E   ;3017  id[fault]_d,d_0
U 125E. 0000.003C.6D80.3C00.0000.125F   ;3018  id[fault]_d
U 125F. 0000.003E.6580.3C00.0000.0001   ;3019  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;3020
```

```
            ;3021  .PAGE
            ;3022  .TOC "   Set Trap Mask"
            ;3023 ;++
            ;3024 ; FUNCTIONAL DESCRIPTION:
            ;3025 ;
            ;3026 ; This routine is used to set a bit in the Micro Trap Mask
            ;3027 ; R[0F].
            ;3028 ;
            ;3029 ; CALLING CONSTRAINT:
            ;3030 ;
            ;3031 ; =0
            ;3032 ;
            ;3033 ; INPUTS:
            ;3034 ;
            ;3035 ; SC - Contains bit number to set.
            ;3036 ;
            ;3037 ; OUTPUTS:
            ;3038 ;
            ;3039 ; rc[0E] - Contains the current trap mask
            ;3040 ;
            ;3041 ; SIDE EFFECTS:
            ;3042 ;
            ;3043 ; d regsiter is destroyed
            ;3044 ;--
            ;3045 SET_TRAP_MASK:
U 1260, 0800,003C,0180,0A78,0000,1261  ;3046  d_r[0f]
U 1261, 0801,001C,0180,0AF8,0000,1262  ;3047  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1262, 0001,003E,0180,0AF0,0000,0001  ;3048  r[0E]_d,return1    ; Save mask and return
            ;3049
```

```
            ;3050  .PAGE
            ;3051  .TOC "   Find MS780-E Memory"
            ;3052 ;++
            ;3053 ; FUNCTIONAL DESCRIPTION:
            ;3054 ;
            ;3055 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;3056 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;3057 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;3058 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;3059 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;3060 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;3061 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;3062 ;     (zero).
            ;3063 ;     All the other MS780-E/H's found will have their starting  address
            ;3064 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;3065 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;3066 ;     apropriate  action.  Register RC[0E] is used as a pointer to the
            ;3067 ;     current MS780-E unit under test.
            ;3068 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;3069 ;     set to maximum.
            ;3070 ;
            ;3071 ; CALLING CONSTRAINT:
            ;3072 ;
            ;3073 ; =00
            ;3074 ;
            ;3075 ; OUTPUTS:
            ;3076 ;
            ;3077 ; rc[0e] - Contains address of Configuration Register
            ;3078 ; r[0]  - Contains TR level
            ;3079 ;
            ;3080 ; SIDE EFFECTS:
            ;3081 ;
            ;3082 ; D register is destroyed.
            ;3083 ;--
            ;3084 =0
            ;3085 FIND.MS780E:
U 10C8, 0000.003D.0180.0800.0000.10C6  ;3086  call[sbi.init]  ; Make sure SBI is enabled
U 10C9, 0003.003C.0180.0988.0000.1263  ;3087  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1263, 0818.0038.1980.0800.0000.1264  ;3088  d_k[zero]  ; Clear MS780-E/H counter
U 1264, 0000.003C.F980.3C00.0000.1265  ;3089  id[3e]_d  ; ...
U 1265, 0018.0038.0580.0A80.0000.1266  ;3090  r[0]_k[.1]  ; Init TR counter
U 1266, 0F03.003C.0180.09F0.0000.10CA  ;3091  d_0,rc[0E]_0  ; Clear "Save" location for
            ;3092      ; ...CNFG A Reg
            ;3093 =0
            ;3094 FIND.MEM.LOOP:
U 10CA, 0800.003D.0180.0A00.0000.128C  ;3095  d_r[0],call[generate.io]; Generate address of TR level
U 10CB, 0001.003C.0180.09F0.0200.10CC  ;3096  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 10CC, 0000.003D.8980.0800.0084.7260  ;3097 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;3098  d[long]_cache.p, ; Read the specified tr level
U 10CD, 0000.003C.0180.C970.0000.1267  ;3099    ic_rc[0e]  ; ...
U 1267, 0000.003C.0180.0A78.0010.1268  ;3100  alu_r[0f],clk.ubcc ; Check for no response
U 1268, 0001.013C.0180.0880.0082.10CE  ;3101  z?,sc_d,ie_ra[0] ; Branch if no adapter here
U 10CE, 0000.003C.0180.0800.0000.1269  ;3102 =0 j/check.adpt.code ; Check for a memory
U 10CF, 0000.003C.0180.0800.0000.1288  ;3103  j/find.mem.lp1  ; Try next tr
            ;3104 CHECK.ADPT.CODE:
```

```
U 1269, 0000,003C,1D80,0800,1404,726A  ;3105  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 126A, 0000,163C,0180,0800,0000,1068  ;3106  state7-4?  ; Branch on the adapter type
U 1068, 0000,003C,8980,0800,0084,726B  ;3107 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1069, 0000,003C,8980,0800,0084,726C  ;3108  j/ms780.c,sc_k[.d] ; MS780-C
U 106A, 0000,003C,0180,0800,0000,1288  ;3109  j/find.mem.lpl  ; Not a memory
U 106B, 0000,003C,0180,0800,0000,1288  ;3110  j/find.mem.lpl  ; Not a memory
U 106C, 0000,003C,6580,0800,0084,7271  ;3111  j/ma780.max,sc_k[.10] ; MA780
U 106D, 0000,003C,0180,0800,0000,1288  ;3112  j/find.mem.lpl  ; Not a memory
U 106E, 0010,0038,0180,09F0,0000,1275  ;3113  rc[0e]_ic,j/found.ms780e ; MS780-E
U 106F, 0010,0038,0180,09F0,0000,1275  ;3114  rc[0e]_ic,j/found.ms780e ; MS780-E
            ;3115 ;
            ;3116 ; Found an MS780-A or -C. Set the starting address
            ;3117 ; to maximum.
            ;3118 ;
U 126B, 0818,0038,5D80,0800,0000,126D  ;3119 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 126C, 0818,0038,0580,0800,0000,126D  ;3120 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;3121 MS780.COMMON:
U 126D, 0819,0030,7180,0800,0000,126E  ;3122  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 126E, 0000,003C,01F8,0803,0080,D26F  ;3123  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 126F, 0D00,003C,0180,0800,0000,1270  ;3124  d_dal.sc  ; Put data in bits<29:14>
            ;3125  cache.p_d[long], ; Set the starting address to maximum
U 1270, 0000,003C,0180,A800,0000,1288  ;3126       j/find.mem.lpl ; and continue TR scan
            ;3127 ;
            ;3128 ; Found an MA780. Set the starting address to maximum
            ;3129 ;
            ;3130 MA780.MAX:
U 1271, 0818,0038,39F8,0803,0000;1272  ;3131  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1272, 0D00,003C,0180,0803,0000,1273  ;3132  d_dal.sc,va_va+4 ; Put in proper position
U 1273, 0000,003C,0180,0803,0000,1274  ;3133  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;3134  cache.p_d[long], ; Set starting address to maximum
U 1274, 0000,003C,0180,A800,0000,1288  ;3135       j/find.mem.lpl
            ;3136 ;
            ;3137 ; Found an MS780E
            ;3138 ;
            ;3139 FOUND.MS780E:
U 1275, 0000,003C,F9F0,2C00,0000,1276  ;3140  q_id[3e]  ; Set up to see how many MS780-E's
            ;3141  alu_q.xor.k[.3], ; Are there Maximum units?
U 1276, 0019,2020,0D80,0800,0010,1277  ;3142   clk.ubcc ; Check condition codes
U 1277, 0000,013C,0180,0800,0000,10D0  ;3143  z?   ; Are we at maximum units for system
U 10D0, 0000,003C,0180,0800,0000,1278  ;3144 =0 J/ms780e.tst  ; No go find how many units ther are
U 10D1, 0818,0038,C180,0800,0000,10D2  ;3145  d_k[.ffff]  ; Yes set address to maximum
U 10D2, 0000,003D,0180,0800,0000,1290  ;3146 =0 call[set.start.addr] ; .....
U 10D3, 0000,003C,0180,0800,0000,1288  ;3147  j/find.mem.lpl  ; Go check to see if we are done
            ;3148 ;
            ;3149 MS780E.TST:
            ;3150  alu_rc[01],  ; Check "Found MS780E Flag"
U 1278, 0010,0038,0180,0908,0010,1279  ;3151   clk.ubcc ; Check condition codes
U 1279, 0810,0138,0180,0970,0000,10D4  ;3152  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;3153 =0 j/not.first.ms780e, ; No
U 10D4, 0818,0038,C180,0800,0000,10D8  ;3154   d_k[.ffff] ; Set starting address
U 10D5, 0001,003C,0180,0988,0000,127A  ;3155  rc[01]_d  ; Yes put base address in rc[01]
U 127A, 0818,0038,7980,0800,0000,127B  ;3156  d_k[.30]  ; Set up SC to point to first unit
U 127B, 0019,0014,F580,0800,0082,127C  ;3157  alu_d+k[.a],sc_alu ; ...
U 127C, 0810,0038,0180,0908,0000,127D  ;3158  d_rc[01]  ; Put base address of unit in D
U 127D, 0000,003C,0180,3400,0000,127E  ;3159  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 127E. 0FD0,003C,0180,0800,0000,10D6  ;3160  d_0   ; Prepare to set starting address to Zero
U 10D6. 0000,003D,0180,0800,0000,1290  ;3161 =0 call[set.start.addr] ; Go do it
          ;3162  j/find.mem.lp1.  ; Check to see if we are done
U 10D7. 0003,003C,0180,09E8,0000,1288  ;3163   rc[0d]_0 ; ....
          ;3164 =0
          ;3165 NOT.FIRST.MS780E:
U 10D8. 0000,003D,0180,0800,0000,1290  ;3166  call[set.start.addr] ; Go set starting address to maximum
U 10D9. 0000,003C,F9F0,2C00,0000,127F  ;3167  q_id[3e]  ; Get the number of MS780-Es found
U 127F. 0819,2014,0580,0800,0000,1280  ;3168  d_q+k[.1]  ; Increment this counter
U 1280. 0000,003C,F980,3C00,0000,1281  ;3169  id[3e]_d  ; Save the counter
U 1281. 0018,0038,11C0,0800,0000,1282  ;3170  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;3171      ; ...were the I/O base addresses will be stored
U 1282. 081D,2000,7980,0800,0000,1283  ;3172  d_q-d,k[.30]  ; ... adjust pointer
U 1283. 0819,0014,7980,0800,0000,1284  ;3173  d_d+k[.30]  ; Point the SC to the ID register
U 1284. 0000,003C,F580,0800,0000,1285  ;3174  k[.a]   ; ...to receive I/O base address
U 1285. 0019,0014,F580,0800,0082,1286  ;3175  alu_d+k[.a],sc_alu ; ...
U 1286. 0810,0038,0180,0970,0000,1287  ;3176  d_rc[0e]  ; Get base address to d
U 1287. 0000,003C,0180,3400,0000,1288  ;3177  id(sc)_d  ; Move base address to ID pointed to by SC
          ;3178
          ;3179 ;
          ;3180 ; Try next tr
          ;3181 ;
          ;3182 FIND.MEM.LP1:
U 1288. 0818,0014,0580,0A80,0000,1289  ;3183  r[0]_la+k[.1],d_alu ; Increment TR level
          ;3184  alu_d.and.k[.f],
U 1289. 0019,0034,6180,0800,0010,128A  ;3185      clk.ubcc  ; Check if all done
U 128A. 0000,013C,0180,0800,0000,10DA  ;3186  z?   ; Branch if yes
U 10DA. 0000,003C,0180,0800,0000,10CA  ;3187 =0 j/find.mem.loop  ; Try next TR
U 10DB. 0810,0038,0180,0908,0010,128B  ;3188  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 128B. 0000,013C,0180,0800,0000,10DC  ;3189  z?   ; Check condition codes
U 10DC. 0001,003E,0180,09F0,0000,0002  ;3190 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10DD. 0000,003E,0180,0800,0000,0001  ;3191  return1   ; No MS780E found
          ;3192
```

```
            ;3193  .PAGE
            ;3194  .TOC "   Generate IO Address"
            ;3195 ;**
            ;3196 ; FUNCTIONAL DESCRIPTION:
            ;3197 ;
            ;3198 ; This routine is used to generate an SBI Configuration Register
            ;3199 ; address from a TR level.
            ;3200 ;
            ;3201 ; CALLING CONSTRAINT:
            ;3202 ;
            ;3203 ; =0
            ;3204 ;
            ;3205 ; INPUTS:
            ;3206 ;
            ;3207 ; D - Contains TR level
            ;3208 ;
            ;3209 ; OUTPUTS:
            ;3210 ;
            ;3211 ; D - Contains SBI IO address
            ;3212 ;--
            ;3213 GENERATE.IO:
U 128C, 0000,003C,89F8,0800,0084,728D  ;3214  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 128D, 0D00,003C,8D80,0800,0084,728E  ;3215  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 128E, 0000,003C,0980,0800,0084,B28F  ;3216  sc_sc-k[.2]  ; insert bit 29
U 128F, 0801,001E,0180,0800,0000,0001  ;3217  d_d.ornot.mask,return1 ; and return
            ;3218
```

```
            ;3219  .PAGE
            ;3220  .TOC "   Set Starting Address"
            ;3221 ;**
            ;3222 ; FUNCTIONAL DESCRIPTION:
            ;3223 ;
            ;3224 ; This routine is used to set the starting address of the
            ;3225 ; memory to the specified value.
            ;3226 ;
            ;3227 ; CALLING CONSTRAINT:
            ;3228 ;
            ;3229 ; =0
            ;3230 ;
            ;3231 ; INPUTS:
            ;3232 ;
            ;3233 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;3234 ;     (B Register Image divided by 2**16)
            ;3235 ; rc[0] - Contains Address of Register A
            ;3236 ;
            ;3237 ; OUTPUTS:
            ;3238 ;
            ;3239 ; d - Contains aligned starting address (B Register Image)
            ;3240 ;
            ;3241 ; SIDE EFFECTS:
            ;3242 ;
            ;3243 ; d,q,va, and sc are destroyed
            ;3244 ;--
            ;3245 SET.START.ADDR:
U 1290, 0010,0038,6580,0970,0284,7291  ;3246  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1291, 0000,003C,01F8,0803,0000,1292  ;3247  va_va*4,q_0  ; Set address to Register B
            ;3248  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1292, 0D00,003C,0980,0800,0084,B293  ;3249       sc_sc-k[.2]  ; Setup to insert bit 14
U 1293, 0801,001C,0180,0800,0000,1294  ;3250  d_d.ornot.mask  ; Insert Write Enable bit
U 1294, 0000,003E,0180,A800,0000,0001  ;3251  cache.p_d[long],return1 ; Set the starting address and return
            ;3252
```

```
            ;3253  .PAGE
            ;3254  .TOC "   Select Controller"
            ;3255 ;**
            ;3256 ; FUNCTIONAL DESCRIPTION:
            ;3257 ;
            ;3258 ; This routine is used to put the memory system into the
            ;3259 ; specified configuration with respect to controller select.
            ;3260 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;3261 ; a RETURN2 is made.
            ;3262 ;
            ;3263 ; CALLING CONSTRAINT:
            ;3264 ;
            ;3265 ; =00
            ;3266 ;
            ;3267 ; INPUTS:
            ;3268 ;
            ;3269 ; d - Contains value to write to bits<2:0> of Register A
            ;3270 ; rc[0e] - Address of Register A
            ;3271 ;
            ;3272 ; SIDE EFFECTS:
            ;3273 ;
            ;3274 ; sc,d,va are destroyed
            ;3275 ;--
            ;3276 SELECT.CNTRLR:
U 1295, 0010.003B.0180.0970.0284.7296  ;3277  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 1296, 0801.001C.0180.0800.0000.1297  ;3278  d_d.ornot.mask  ; Insert the enable bit into D reg
U 1297, 0000.003C.0180.A800.0000.1298  ;3279  cache.p_d[long]  ; Write the configuration Register
U 1298, 0818.003B.5D80.0800.0000.1299  ;3280  d_k[.7]   ; Get Misconfiguration bits
U 1299, 0000.003C.61F8.0800.0084.729A  ;3281  sc_k[.F],q_0  ; ...
U 129A, 0D00.003C.0180.0800.0000.129B  ;3282  d_dal.sc  ; ... D = x38000
U 129B, 0000.003C.01E0.0800.0000.129C  ;3283  q_d   ; Save mask
U 129C, 0000.003C.0180.C800.0000.129D  ;3284  d[long]_cache.p  ; Read CNFG A Reg and
            ;3285  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 129D, 001D.2034.0180.0800.0010.129E  ;3286   clk.ubcc ; ...
U 129E, 0000.013C.0180.0800.0000.10DE  ;3287  z?   ; Branch if no
U 10DE, 0000.003E.0180.0800.0000.0001  ;3288 =0 RETURN1   ; Bad configuration
U 10DF, 0000.003E.0180.0800.0000.0002  ;3289  RETURN2   ; Configuration ok
            ;3290
```

```
            ;3291  .PAGE
            ;3292  .TOC "   SELECT LOWER CONTROLLER"
            ;3293 ;*****************************************************************
            ;3294 ;**
            ;3295 ;
            ;3296 ; Functional Description:
            ;3297 ; ----------------------
            ;3298 ;
            ;3299 ; This subroutine can be called to select the lower
            ;3300 ; Controller on a MS780-E/H.
            ;3301 ; If the Lower controller is misconfigured then a
            ;3302 ; RETURN1 will be made.  Otherwise a RETURN2
            ;3303 ;
            ;3304 ; Calling Constraint: =00
            ;3305 ; ------------------
            ;3306 ;
            ;3307 ;
            ;3308 ; Inputs:
            ;3309 ; ------
            ;3310 ;
            ;3311 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;3312 ;
            ;3313 ; Outputs:
            ;3314 ; -------
            ;3315 ;
            ;3316 ; RC[0D] will have the address of CNFG Register C
            ;3317 ;        ( 2000x008 )
            ;3318 ;
            ;3319 ; Side Effects:
            ;3320 ; ------------
            ;3321 ;
            ;3322 ; Contents of the following registers will lost:
            ;3323 ;
            ;3324 ; D
            ;3325 ;
            ;3326 ; The Lower controller on the MS780-E/H will be configured
            ;3327 ; when the subroutine is exitted with a RETURN2
            ;3328 ;--
            ;3329 ;*****************************************************************
            ;3330 =00
            ;3331 SELECT.LOWER:
            ;3332  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1008, 0818,0039,1980,0800,0000,1295  ;3333   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1009, 0000,003E,0180,0800,0000,0001  ;3334  RETURN1    ; Lower Controller Misconfigured
U 100A, 0810,0038,0180,0970,0000,100B  ;3335  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;3336  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 100B, 0019,0016,0180,09E8,0000,0002  ;3337   RETURN2    ; Let caller know that the controller
            ;3338       ; ...was configured properly
            ;3339
```

```
            ;3340  .PAGE
            ;3341  .TOC "   SELECT UPPER CONTROLLER"
            ;3342 ;*************************************************************
            ;3343 ;**
            ;3344 ;
            ;3345 ; Functional Description:
            ;3346 ; ----------------------
            ;3347 ;
            ;3348 ; This subroutine can be called to select the upper
            ;3349 ; Controller on a MS780-E/H.
            ;3350 ; If the Upper controller is misconfigured then a
            ;3351 ; RETURN1 will be made.  Otherwise a RETURN2
            ;3352 ;
            ;3353 ; Calling Constraint: =00
            ;3354 ; ------------------
            ;3355 ;
            ;3356 ;
            ;3357 ; Inputs:
            ;3358 ; ------
            ;3359 ;
            ;3360 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;3361 ;
            ;3362 ; Outputs:
            ;3363 ; -------
            ;3364 ;
            ;3365 ; RC[0D] will have the address of CNFG Register D
            ;3366 ;        ( 2000x010 )
            ;3367 ;
            ;3368 ; Side Effects:
            ;3369 ; ------------
            ;3370 ;
            ;3371 ; Contents of the following registers will lost:
            ;3372 ;
            ;3373 ; D
            ;3374 ;
            ;3375 ; The Upper controller on the MS780-E/H will be configured
            ;3376 ; when the subroutine is exitted with a RETURN2
            ;3377 ;--
            ;3378 ;*************************************************************
            ;3379 =00
            ;3380 SELECT.UPPER:
            ;3381  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1010, 0818,0039,0980,0800,0000,1295  ;3382   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1011, 0000,003E,0180,0800,0000,0001  ;3383  RETURN1    ; Upper Controller Misconfigured
U 1012, 0810,0038,0180,0970,0000,1013  ;3384  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;3385  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1013, 0019,0016,8580,09E8,0000,0002  ;3386   RETURN2   ; Let caller know that the controller
            ;3387       ; ...was configured properly
            ;3388
```

```
          ;3389  .PAGE
          ;3390  .TOC "   ENABLE NEXT MS780-E"
          ;3391 ;++
          ;3392 ; FUNCTIONAL DESCRIPTION:
          ;3393 ;
          ;3394 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
          ;3395 ;    They will be tested  separately, according to the TR level at which
          ;3396 ;    they are located,  starting with the one  at the lowest level first.
          ;3397 ;          When  a test has finished  with the first unit, it will have to call
          ;3398 ;    this specific  routine  to see  if any other MS780-E unit is present
          ;3399 ;    on the SBI. If more units are  present, the first one  will be  dis-
          ;3400 ;    abled  by  setting  its  starting  address to maximum, the next unit
          ;3401 ;    will be  enabled  by  setting its starting address to 0 and the test
          ;3402 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
          ;3403 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
          ;3404 ;    ID registers 3A through 3D and  a count of the Total number of units
          ;3405 ;    found - 1 in register 3E. In this subroutine the SC register is used
          ;3406 ;    as  a  pointer  to  ID  registers 3A through 3D.
          ;3407 ;
          ;3408 ; CALLING CONSTRAINT:
          ;3409 ;
          ;3410 ;  =00
          ;3411 ;
          ;3412 ;--
          ;3413 ENABLE.NEXT.MS780E:
U 129F, 0000,003C,F9F0,2C00,0000,12A0  ;3414  Q_ID[3E]  ; Get total number of MS780E to Q
          ;3415  ALU_Q,    ; Check to see if it is zero
U 12A0, 0001,203C,0180,0800,0010,12A1  ;3416     CLK.UBCC  ; Check Condition Codes
U 12A1, 0000,013C,0180,0800,0000,10E0  ;3417  Z?   ; ...for zero
U 10E0, 0000,003C,0180,0800,0000,12A2  ;3418 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10E1, 0000,003E,0180,0800,0000,0002  ;3419  RETURN2   ; Zero No more units to test
          ;3420 ENABLE.NEXT:
U 12A2, 0818,0038,C180,0800,0000,10E2  ;3421  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10E2, 0000,003D,0180,0800,0000,1290  ;3422 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10E3, 0818,0038,7980,0800,0000,12A3  ;3423  D_K[.30]  ; Prepare to point to the ID register holding
          ;3424     ; ...the I/O Base address of the next MS780-E
U 12A3, 0819,0014,1180,0800,0000,12A4  ;3425  D_D+K[.4]  ; ...
U 12A4, 0000,003C,F580,0800,0000,12A5  ;3426  K[.A]   ; ...
U 12A5, 0819,0014,F580,0800,0000,12A6  ;3427  D_D+K[.A]  ; ...
U 12A6, 0000,003C,F9F0,2C00,0000,12A7  ;3428  Q_ID[3E]  ; Get MS780-E Counter
          ;3429  D_D-Q,    ; D now holds the pointer to the ID register
U 12A7, 081D,0000,0180,0800,0082,12A8  ;3430   SC_ALU  ; ...
U 12A8, 0000,003C,01F0,2400,0000,12A9  ;3431  Q_ID(SC)  ; Q gets adrress of next MS780E
U 12A9, 0001,203C,0180,09F0,0000,12AB  ;3432  RC[0E]_Q  ; Save it also in RC[0E]
U 12AB, 0F03,003C,0180,09E8,0000,10E4  ;3433  RC[0D]_0,D_0  ; Set starting address to zero
U 10E4, 0000,003D,0180,0800,0000,1290  ;3434 =0 CALL[SET.START.ADDR] ; ....
U 10E5, 0F00,003C,F9F0,2C00,0000,12AC  ;3435  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 12AC, 081D,2008,0180,0800,0000,12AD  ;3436  D_Q-D-1   ; Subtract 1 from total
U 12AD, 0000,003C,F980,3C00,0000,10E6  ;3437  ID[3E]_D  ; Adjust the pointer
U 10E6, 0000,003D,0180,0800,0000,125B  ;3438 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10E7, 0000,003E,0180,0800,0000,0001  ;3439  RETURN1   ; Tell caller another unit was found
          ;3440
```

```
            ;3441  .PAGE
            ;3442  .TOC "   CONFIGURE MS780-E/H"
            ;3443 ;*************************************************************
            ;3444 ;**
            ;3445 ;
            ;3446 ; Functional Description:
            ;3447 ; ----------------------
            ;3448 ;
            ;3449 ;       This  subroutine  will  be  used  by  tests  that   do   not
            ;3450 ; specifically  check  the  memory, but  make use of it to execute.
            ;3451 ; Its purpose is to find a MS780-E and  configure  it  properly  so
            ;3452 ; that  if  a  Read/Write operation will take place no false errors
            ;3453 ; will be logged due to misconfigured memory.
            ;3454 ;
            ;3455 ; Calling Constraint: =00
            ;3456 ; ------------------
            ;3457 ;
            ;3458 ;
            ;3459 ; Inputs:
            ;3460 ; ------
            ;3461 ;
            ;3462 ; RC[0E] - I/O BASE OF MS780-E/H
            ;3463 ;
            ;3464 ;
            ;3465 ; Outputs:
            ;3466 ; -------
            ;3467 ;
            ;3468 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
            ;3469 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
            ;3470 ;       OR IF NO MS780-E/Hs WERE FOUND
            ;3471 ;       ON THE SBI
            ;3472 ;
            ;3473 ; Side Effects:
            ;3474 ; ------------
            ;3475 ;
            ;3476 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
            ;3477 ;
            ;3478 ;
            ;3479 ;
            ;3480 ;--
            ;3481 ;*************************************************************
            ;3482 CONFIG.MS780E:
U 12AE, 0818,0038,0D80,0800,0000,12AF  ;3483  D_K[.3]     ; Set up flag for the Fail Chain
U 12AF, 0001,003C,0180,09E8,0000,1014  ;3484  RC[0D]_D     ; ...
U 1014, 0000,003D,0180,0800,0000,10C8  ;3485 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1015, 0000,003D,0980,0800,0084,7024  ;3486  ERROR1[.2]    ; No MS780-E/H's found on the SBI
U 1016, 0000,003C,0180,0800,0000,1050  ;3487  NOP     ; OK. Found at least one MS780-E/H
            ;3488 =
            ;3489 CONFIG.RETRY:     ; Entry point for the case in which the
            ;3490       ; ...first MS780-E/H could not be
            ;3491       ; ...properly configured
U 1050, 0000,003D,0180,0800,0000,1008  ;3492 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1051, 0000,003C,0180,0800,0000,1054  ;3493  J/CNFG.LMIS    ; Lower controller misconfigured
U 1052, 0000,003E,0180,0800,0000,0001  ;3494  RETURN1     ; OK. Lower Controller is configured
            ;3495 =
```

```
            ;3496 =00
            ;3497 CNFG.LMIS:
U 1054. 0000.003D.0180.0800.0000.1010  ;3498  CALL[SELECT.UPPER]  ; Select the upper controller
U 1055. 0000.003C.0180.0800.0000.1060  ;3499  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 1056. 0000.003E.0180.0800.0000.0001  ;3500  RETURN1    ; OK. Upper Controller is configured
            ;3501 =
            ;3502 =00
            ;3503 CNFG.UMIS:
U 1060. 0000.003D.0180.0800.0000.129F  ;3504  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;3505       ; ...MS780-E/H found so go and see
            ;3506       ; ...if there are any more
U 1061. 0000.003C.0180.0800.0000.12B0  ;3507  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;3508       ; ...try to configure it
U 1062. 0818.0038.0D80.0800.0000.1063  ;3509  D_K[.3]    ; Set up flag for the Fail Chain
U 1063. 0001.003C.0180.09E8.0000.12B0  ;3510  RC[0D]_D    ; ...
U 12B0. 0000.003D.0980.0800.0084.7024  ;3511  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;3512       ; ...abort the test
            ;3513 =
            ;3514
```

```
            ;3515  .PAGE
            ;3516  .TOC "    GET MEMORY SIZE"
            ;3517 ;*************************************************************
            ;3518 ;**
            ;3519 ;
            ;3520 ; Functional Description:
            ;3521 ; ----------------------
            ;3522 ;
            ;3523 ; This subroutine can be used to get the size of the memory.
            ;3524 ; CNFG Reg A Bits <14:09> have the memory size in 1 Meg
            ;3525 ; increments.
            ;3526 ;
            ;3527 ; Calling Constraint: =0
            ;3528 ; ------------------
            ;3529 ;
            ;3530 ;
            ;3531 ; Inputs:
            ;3532 ; ------
            ;3533 ;
            ;3534 ; RC[0E] Must have the Address of CNFG Register A
            ;3535 ;
            ;3536 ; Outputs:
            ;3537 ; -------
            ;3538 ;
            ;3539 ; Register D will contain upon exit the size of the
            ;3540 ; memory aligned on Mega-byte boundary.
            ;3541 ;
            ;3542 ; Side Effects:
            ;3543 ; ------------
            ;3544 ;
            ;3545 ; The contents of the following registers will be lost:
            ;3546 ;
            ;3547 ; D, SC, Q
            ;3548 ;
            ;3549 ;--
            ;3550 ;*************************************************************
            ;3551 GET.MEMORY.SIZE:
U 12B1, 0010,0038,0180,0970,0200,12B2  ;3552  VA_RC[0E]    ; Point to CNFG Reg A
U 12B2, 0000,003C,0180,C800,0000,12B3  ;3553  D[LONG]_CACHE.P    ; Read the register
U 12B3, 001B,0000,D980,0800,0082,12B4  ;3554  ALU_0-K[.9],SC_ALU  ; Get -10 in the SC
U 12B4, 0D00,003C,0180,0800,0000,12B5  ;3555  D_DAL.SC    ; Shift bits <14:9> to <5:0>
U 12B5, 0819,0034,5580,0800,0000,1064  ;3556  D_D.AND.K[.3F]    ; Mask unwanted bits and clear Reg Q
            ;3557 =00 CALL[64K.OR.256K],   ; Find out the size of the arrays
U 1064, 0000,003D,01E0,0800,0000,12B8  ;3558   Q_D    ; Save Memory size bits in Q
U 1065, 0000,003D,0D80,0800,0084,7024  ;3559  ERROR1[.3]    ; RET1 Log an Error. Mixed or Missing
            ;3560       ; ...arrays on the controller
            ;3561 ;
            ;3562 ; The duplication of the next two return status is necessary so the returning
            ;3563 ; subroutine (64k.or.256k) can be used in other modules without any
            ;3564 ; modifications.
            ;3565 ;
U 1066, 0819,2014,0580,0800,0000,1067  ;3566  D_Q+K[.1]    ; RET2 Increment memory size by one
            ;3567       ; ...(found 1 Meg arrays)
U 1067, 0819,2014,0580,0800,0000,12B6  ;3568  D_Q+K[.1]    ; RET3 Increment memory size by 1
            ;3569       ; ...(found 4 Meg arrays)
```

```
U 12B6, 0000,003C,21F8,0800,0084,72B7  ;3570  SC_K[.14],Q_0   ; Prepare to shift to Megabyte position
          ;3571  D_DAL.SC,   ; Shift bits <5:0> to <25:20>
U 12B7, 0D00,003E,0180,0800,0000,0001  ;3572   RETURN1   ; Return with memory size in Register D
          ;3573
```

```
                ;3574  .PAGE
                ;3575  .TOC "    64K OR 256K CHIPS"
                ;3576 ;*************************************************************
                ;3577 ;**
                ;3578 ;
                ;3579 ; Functional Description:
                ;3580 ; ----------------------
                ;3581 ; This subroutine is used to find the type of chips
                ;3582 ; on the MS780-E/H arrays.  The RETURN modes are as
                ;3583 ; follows:
                ;3584 ;
                ;3585 ; RETURN1 = Error. Mixed 64K and 256K arrays
                ;3586 ;
                ;3587 ; RETURN2 = 64K chips on the array
                ;3588 ;
                ;3589 ; RETURN3 = 256K chips on the array
                ;3590 ;
                ;3591 ; Calling Constraint: =00
                ;3592 ; ------------------
                ;3593 ;
                ;3594 ;
                ;3595 ; Inputs:
                ;3596 ; ------
                ;3597 ;
                ;3598 ; RC[0E] must hold the I/O base address of the MS780-E/H
                ;3599 ; currently under test
                ;3600 ;
                ;3601 ; Outputs:
                ;3602 ; -------
                ;3603 ;
                ;3604 ; NO specific outputs. (See RETURN modes above)
                ;3605 ;
                ;3606 ;
                ;3607 ; Side Effects:
                ;3608 ; ------------
                ;3609 ;
                ;3610 ; The contents of the following registers will be lost:
                ;3611 ;
                ;3612 ; D
                ;3613 ;
                ;3614 ;--
                ;3615 ;*************************************************************
                ;3616 64K.OR.256K:
U 12B8, 0010,0038,0180,0970,0200,12B9  ;3617  VA_RC[0E]    ; Point to CNFG Register A
U 12B9, 0000,003C,0180,C800,0000,12BA  ;3618  D[LONG]_CACHE.P    ; Read the register
U 12BA, 0200,003C,0180,0800,0000,12BB  ;3619  D_D.RIGHT2    ; Shift received contents two bits
                ;3620          ; ...to the right
U 12BB, 0600,003C,0180,0800,0000,12BC  ;3621  D_D.RIGHT    ; Shift one more time
U 12BC, 0000,193C,0180,0800,0000,107C  ;3622  D1-0?    ; Branch on the RAM Type
U 107C, 0000,003E,0180,0800,0000,0001  ;3623 =1100 RETURN1    ; Error: both 64K and 256K RAMs present
U 107D, 0000,003E,0180,0800,0000,0002  ;3624  RETURN2    ; 64K RAMs found
U 107E, 0000,003E,0180,0800,0000,0003  ;3625  RETURN3    ; 256K RAMs found
U 107F, 0000,003E,0180,0800,0000,0001  ;3626  RETURN1    ; Error: missing arrays
                ;3627 =
```

```
;3628 .PAGE "TEST 1FB SBI/CACHE MISCELLANEOUS TESTS"
;3629 ;=============================================================================
;3630 ;**
;3631 ;
;3632 ; SBI/CACHE MISCELLANEOUS TESTS
;3633 ;
;3634 ; TEST DESCRIPTION
;3635 ;
;3636 ; THIS TEST CHECK THAT ON READ MISSES THE SBI (SBH,SBL)
;3637 ; DRIVE THE PROPER ADDRESS ON TO THE PA BUS.  THE TEST
;3638 ; CHECKS ALL ADDRESS LINES (BYTE ADDRESS BIT 2 TO 28).  THE LINES
;3639 ; ARE CHECKED TO SEE THAT THEY ARE NOT STUCK AND ARE
;3640 ; INDEPENDENT.  THE TEST USES THE HIT/MISS LOGIC TO
;3641 ; SEE THAT THE INTERFACE DRIVES THE CORRECT ADDRESS TO THE
;3642 ; MEMORY SPACE, AND THE OTHER TEST CHECKS THE ADDRESS
;3643 ; LINES OUTSIDE OF VALID MEMORY SPACE. ALSO CHECKED IS STALL IN THE
;3644 ; PRESENTS OF CACHE PARITY ERRORS.
;3645 ;
;3646 ;
;3647 ; SUBTEST 1 TEST ALL ADDRESS LINES WITHIN VALID MEMORY
;3648 ;   SPACE ARE INDEPENDENT.  SETS STARTING ADDRESS
;3649 ;   TO ZERO AND RUNS THE TEST.
;3650 ;
;3651 ; SUBTEST 2 TEST THE ADDRESS LINES ABOVE VALID
;3652 ;   MEMORY SPACE.  THE MEMORY STARTING ADDRESS
;3653 ;   IS DISPLACE TO THE BIT UNDER TEST AND
;3654 ;   THE TEST IS RUN.
;3655 ;
;3656 ; SUBTEST 3 CHECK NO STALL(I.E. NO SBI CYCLES) WHEN A READ
;3657 ;   MISS AND CACHE PARITY EREOS EXSISTS(BOTH GROUPS).
;3658 ;
;3659 ; SUBTEST 4 CHECK STALL( I.E. SBI CYCLES) WHEN A READ MISS
;3660 ;   OCCURES WITH A FORCE MISS AND PARITY ERRORS(BOTH
;3661 ;   GROUPS).
;3662 ;
;3663 ;
;3664 ; LOGIC DESCRIPTION
;3665 ;
;3666 ;  THE LOGIC TO DRIVE THE PA BUS ON THE SBH (M8219),SBL
;3667 ;  (M8218) BOARDS.
;3668 ;
;3669 ; ERROR LOGOUT
;3670 ;
;3671 ; SUBTEST 2,1  ==============================================
;3672 ; DATA: TEST LOCATION (ADDRESS BIT BEING TESTED)
;3673 ;  GROUP FLAG (0=GROUP0, 1=GROUP1)
;3674 ;  MAXIMUM ADDRESS FROM CONFIGURATION REG A
;3675 ;  TR LEVEL OF MEMORY BEING USED; <15>=1=16K CTRLR FLAG
;3676 ;  UNDEFINED
;3677 ;  UNDEFINED
;3678 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3679 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3680 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3681 ;  INTERRUPT FLAG:
;3682 ;   0=>NO INTERRUPT OCCURRED
```

```
            ;3683 ;   1=>AN INTERRUPT OCCURRED
            ;3684 ;
            ;3685 ; SUBTEST 3,4  ================================================
            ;3686 ;   SEE SUBTEST FOR DESCRIPTION
            ;3687 ; SYNC POINT DESCRIPTION
            ;3688 ;
            ;3689 ;--
            ;3690 ;
            ;3691 ;=============================================================================
            ;3692 ;  SCRATCH PADS
            ;3693 ; RA
            ;3694 ; 0 HIT/MISS MASK     ^X 600
            ;3695 ; 8 MAX ADDRESS FROM CONFIGURATION REG A
            ;3696 ; 9 1ST ADDRESS BIT ABOVE VALID MAMORY
            ;3697 ; F U-TRAP FLAG
            ;3698 ;
            ;3699 ; RC
            ;3700 ; 0 CONFIG REG A ADDRESS  ^X 2000 2000
            ;3701 ; 1 CONFIG REG B ADDRESS  ^X 2000 2004
            ;3702 ; 2 CONFIG REG C ADDRESS  ^X 2000 2008
            ;3703 ; F # ARG TO CONSOLE
            ;3704 ;
            ;3705 ;=============================================================================
            ;3706 ;
            ;3707 =0
U 10E8, 0000,003D,0180,0800,0000,103A  ;3708 T1CE: NEWTST
U 10E9, 0000,003C,0180,0800,0000,10EA  ;3709  NOP
U 10EA, 0000,003D,0180,0800,0000,12AE  ;3710 =0 CALL[CONFIG.MS780E]  ; Find MS780E's on the SBI
            ;3711       ; ...and configure them
U 10EB, 0010,0038,01C0,0970,0000,12BD  ;3712  Q_RC[0E]   ; Save IO Base address of the MS780E
U 12BD, 0F00,003C,0180,0800,0000,12BE  ;3713  D_0    ;
U 12BE, 0000,003C,7180,3C00,0000,12BF  ;3714  ID[COMP]_D
U 12BF, 0001,203C,0180,09D0,0000,12C0  ;3715  RC[0A]_Q
U 12C0, 0018,0038,F580,09F8,0000,12C1  ;3716  RC[0F]_K[.A]
U 12C1, 0818,0038,D5F8,0800,0000,12C2  ;3717  D_K[.6],Q_0   ;RA 0
U 12C2, 0000,003C,0180,0800,0084,72C3  ;3718  SC_K[.8]
U 12C3, 0D00,003C,0180,0800,0000,12C4  ;3719  D_DAL.SC
U 12C4, 0001,003C,0180,0A80,0000,10EC  ;3720  R[0]_D
            ;3721  ;=============================================================================
            ;3722  ;
U 10EC, 0000,003D,0180,0800,0000,12B1  ;3723 =0 CALL[GET.MEMORY.SIZE]  ; Get the size of the memory
            ;3724       ; ...on the MS780-E
U 10ED, 0001,003C,0180,0AC0,0000,12C5  ;3725  R[8]_D    ; Save in RA[8]
U 12C5, 0001,003C,0180,09E0,0000,10EE  ;3726  RC[0C]_D   ; Save also in RC[0C]
            ;3727 ;=============================================================================
            ;3728 ;*
            ;3729 ;SUBTEST 1
            ;3730 ;
            ;3731 ; ADDRESS LINES WITHIN VALID MEMORY SPACE
            ;3732 ; CHECK TO SEE THAT THE SBI INTERFACE (SBL,SBH)
            ;3733 ; DRIVE THE PA BUS.  THE TEST CHECKS THAT ALL ADDRESS
            ;3734 ;  LINES ARE NOT STUCK AND ARE INDEPENDENT.
            ;3735 ;
            ;3736 ;    SEQUENCE OF EVENTS
            ;3737 ;
```

```
                ;3738 ; 1)  INVALIDATE ALL OF CACHE
                ;3739 ;  AND FORCE TO ONE GROUP
                ;3740 ; 2)  INVALIDATE LOCATION (QUAD) 0
                ;3741 ; 3)  INVALIDATE TEST LOCATION
                ;3742 ; 4)  READ TEST LOCATION
                ;3743 ;  SEE MISS
                ;3744 ;  IF HIT:  FAILURE IN HIT/MISS LOGIC
                ;3745 ; 5)  READ LOCATION 0 (EXCEPT IF ADDRESS(TEST) IS ABOVE 1000)
                ;3746 ;  SEE MISS
                ;3747 ;  IF HIT:  TEST LOCATION ADDRESS BIT
                ;3748 ;   DID NOT TOGGLE(SBI DID NOT DRIVE
                ;3749 ;   THE ADDRESS BIT OVER THE PA BUS)
                ;3750 ; 6)  READ TEST LOCATION
                ;3751 ;  SEE HIT
                ;3752 ;  IF MISS: TEST ADDRESS BIT TIE TO
                ;3753 ;   A ANOTHER ADDRESS BIT ON
                ;3754 ;   SBL.SBH(SBI INERFACE DID NOT DRIVE
                ;3755 ;   THE CORRECT ADDRESS(S) BITS OVER THE PA BUS)
                ;3756 ;
                ;3757 ; 7)  INVALIDATE LOCATION 0, TEST LOCATION
                ;3758 ; 8)  REPEAT STEP 2-7 UNTILL MAX ADDRESS IS
                ;3759 ;  REACHED
                ;3760 ;
                ;3761 ;ERROR LOGOUT
                ;3762 ;
                ;3763 ; DATA: TEST ADDRESS (FAILING ADDRESS)
                ;3764 ;  GROUP UNDER TEST (0=GROUP0,1=GROUP1)
                ;3765 ;  MAXIMUM ADDRESS
                ;3766 ;  UNDEFINED
                ;3767 ;  UNDEFINED
                ;3768 ;  UNDEFINED
                ;3769 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
                ;3770 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
                ;3771 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;3772 ;  INTERRUPT FLAG:
                ;3773 ;   0=>NO INTERRUPT OCCURRED
                ;3774 ;   1=>AN INTERRUPT OCCURRED
                ;3775 ;-
                ;3776 ;================================================================================
                ;3777 ;
U 10EE, 0000,003D,0180,0800,0000,11B9  ;3778 =0 CALL[SUBTEST1]
U 10EF, 0000,003C,0180,0800,0000,10F0  ;3779  NOP
U 10F0, 0000,003D,0180,0800,0000,11DB  ;3780 =0 CALL[INIT]
U 10F1, 0003,003C,0180,09E8,0000,12C6  ;3781  RC[0D]_0    ;SET GROUP FLAG
U 12C6, 0003,003C,0180,09F0,0000,10F2  ;3782 SXX010: RC[0E]_0    ;INVALIDATE ALL OF CACHE
U 10F2, 0000,003D,0180,0800,0000,11D0  ;3783 =0 CALL[INVALIDATECACHE]  ; INVALIDATE THE GROUP UNDER TEST
U 10F3, 001B,0038,0180,09F0,0000,10F4  ;3784 SXX030: RC[0E]_K[.8]    ;SET INITIAL ADDRESS FOR TEST
U 10F4, 0000,003D,0180,0800,0000,119C  ;3785 =0 ERLOOP
U 10F5, 0000,003C,0180,0800,0000,10F6  ;3786 SXX040: NOP
U 10F6, 0000,003D,0180,0800,0000,11DB  ;3787 =0 CALL[INIT]
U 10F7, 081B,0038,4580,0800,0000,12C7  ;3788  D_K[.8000]    ;SET GROUP UNDER TEST
U 12C7, 0010,0038,0180,0968,0010,12C8  ;3789  ALU_RC[0D],CLK.UBCC
U 12C8, 0600,013C,0180,0800,0000,10F8  ;3790  Z?,D_D.RIGHT    ;GROUP 0 UNDER TEST
U 10F8, 0600,003C,0180,0800,0000,10F9  ;3791 =0 D_D.RIGHT    ;GROUP 1 UNDER TEST
U 10F9, 0000,003C,7580,3C00,0000,12C9  ;3792  ID[MAINT]_D    ;WRITE THE I.D. REGISTER
```

```
U 12C9, 0018.0038.1980.0800.0200.12CA  ;3793  VA_K[ZERO]    ;INVALIDATE LOCATION 0(CACHE)
U 12CA, 0000.003C.0180.9000.0000.12CB  ;3794  MCT/INVALIDATE
U 12CB, 0010.0038.0180.0970.0200.12CC  ;3795  VA_RC[0E]    ;INVALIDATE TEST LOCATION(CACHE)
U 12CC, 0000.003C.0180.9000.0000.12CD  ;3796  MCT/INVALIDATE
U 12CD, 0010.0038.0180.0970.0200.12CE  ;3797  VA_RC[0E]    ;READ TEST LOCATION FOR MISS
U 12CE, 0000.003C.0180.C800.0000.1070  ;3798  MCT/READ.P    ; 1ST READ
U 1070, 0900.003D.0180.0800.0000.1030  ;3799 =00 CALL[NOINTR]
U 1071, 0000.003D.0180.0800.0000.119D  ;3800   ERROR1     ;UNEXPECTED INTERRUPT
U 1072, 0000.003C.75F0.2C00.0000.1073  ;3801  Q_ID[MAINT]    ;CHECK FOR A MISS
U 1073, 000D.2034.0180.0A00.0010.12CF  ;3802  ALU_Q[AND]R[0],CLK.UBCC
U 12CF, 0000.013C.0180.0800.0000.10FA  ;3803  Z?
U 10FA, 0000.003D.0180.0800.0000.119D  ;3804 =0 ERROR1     ;GOT A HIT,TEST ADDRESS
          ;3805       ;  LOCATION NOT INVALIDATED
U 10FB, 0818.0038.6D80.0800.0000.12D0  ;3806  D_K[.FFF0]    ;ADDRESS ABOVE 1000
U 12D0, 0200.003C.0180.0800.0000.12D1  ;3807  D_D.RIGHT2
U 12D1, 0200.003C.0180.0800.0000.12D2  ;3808  D_D.RIGHT2
U 12D2, 0011.0034.0180.0970.0010.12D3  ;3809  ALU_D[AND]RC[0E],CLK.UBCC ;ABOVE 1000 ??
U 12D3, 0000.013C.0180.0800.0000.10FC  ;3810  Z?
U 10FC, 0000.003C.0180.0800.0000.12D4  ;3811 =0 J/SXX043    ;BELOW 1000, READ LOCATION 0
U 10FD, 0000.003C.0180.0800.0000.10FF  ;3812  J/SXX047    ;ABOVE 1000,SKIP READING LOC 0
U 12D4, 0018.0038.1980.0800.0200.12D5  ;3813 SXX043: VA_K[ZERO]    ;READ LOCATION ZERO(0)
U 12D5, 0000.003C.0180.C800.0000.1074  ;3814  MCT/READ.P    ;  FOR A MISS
U 1074, 0000.003D.0180.0800.0000.1030  ;3815 =00 CALL[NOINTR]
U 1075, 0000.003D.0180.0800.0000.119D  ;3816  ERROR1     ;UNEXPECTED INTERRUPT!!!!
U 1076, 0000.003C.75F0.2C00.0000.1077  ;3817  Q_ID[MAINT]
U 1077, 000D.2034.0180.0A00.0010.12D6  ;3818  ALU_Q[AND]R[0],CLK.UBCC
U 12D6, 0000.013C.0180.0800.0000.10FE  ;3819  Z?
U 10FE, 0000.003D.0180.0800.0000.119D  ;3820 =0 ERROR1     ;GOT A HIT,SBI INTERFACE
          ;3821       ;  DID NOT DRIVE ADDRESS BIT
          ;3822       ;  ON THE PA BUS
U 10FF, 0010.0038.0180.0970.0200.12D7  ;3823 SXX047: VA_RC[0E]    ;READ TEST ADDRESS FOR HIT
U 12D7, 0000.003C.0180.C800.0000.1078  ;3824  MCT/READ.P    ; 2ND READ
U 1078, 0000.003D.0180.0800.0000.1030  ;3825 =00 CALL[NOINTR]
U 1079, 0000.003D.0180.0800.0000.119D  ;3826  ERROR1     ;UNEXPECTED INTERRUPT !!!!!!!
U 107A, 0000.003C.75F0.2C00.0000.107B  ;3827  Q_ID[MAINT]
U 107B, 000D.2034.0180.0A00.0010.12D8  ;3828  ALU_Q[AND]R[0],CLK.UBCC  ;DID A HIT OCCURR
U 12D8, 0000.013C.0180.0800.0000.110A  ;3829  Z?
U 110A, 0000.003C.0180.0800.0000.12D9  ;3830 =0 J/SXX050    ;GREAT, GOT THE HIT
U 110B, 0000.003D.0180.0800.0000.119D  ;3831  ERROR1     ;SBI INTERFACE DID NOT DRIVE
          ;3832       ;  THE COORRECT BIT ONTO THE
          ;3833       ;  PA BUS
U 12D9, 0810.0038.0180.0970.0000.12DA  ;3834 SXX050: D_RC[0E]    ;FIND NEXT ADDRESS
U 12DA, 0500.003C.01C0.0A40.0000.12DB  ;3835  D_D.LEFT,Q_R[8]    ;SHIFT,GET MAX ADDRESS
U 12DB, 001D.0000.0180.0800.0010.12DC  ;3836  ALU_D[A-B]Q,CLK.UBCC  ;COMPARE,CURRENT-MAX
U 12DC, 0001.1B3C.0180.09F0.0000.1043  ;3837  ALU?,RC[0E]_D
U 1043, 0000.003C.0180.0800.0000.12DD  ;3838 =0011 J/SXX060    ;00: DONE
U 1047, 0000.003C.0180.0800.0000.12DD  ;3839  J/SXX060    ;01: DONE
U 104B, 0000.003C.0180.0800.0000.10F5  ;3840  J/SXX040    ;10: NOT DONE,REPEAT
U 104F, 0000.003C.0180.0800.0000.12DD  ;3841  J/SXX060    ;11: DONE
U 12DD, 0010.0038.0180.0968.0010.12DE  ;3842 SXX060: ALU_RC[0D],CLK.UBCC  ;CHECK FOR 2ND GROUP DONE
U 12DE, 0818.0138.4580.0800.0000.1110  ;3843  Z?,D_K[.8000]
U 1110, 0000.003C.0180.0800.0000.12E0  ;3844 =0 J/SXX070    ;DONE  !
U 1111, 0218.0038.0580.09E8.0000.12DF  ;3845  D_D.RIGHT2,RC[0D]_K[.1]
U 12DF, 0000.003C.7580.3C00.0000.12C6  ;3846  ID[MAINT]_D,J/SXX010  ;REPEAT SUBTEST WITH OTHER GROUP
U 12E0, 0000.003C.0180.0800.0000.1112  ;3847 SXX070: NOP
```

```
;3848 ;
;3849 ;*****************************************************************************
;3850 ;*
;3851 ;SUBTEST 2
;3852 ;
;3853 ; ADDRESS LINES ABOVE THE VALID MEMORY SPACE.  THIS
;3854 ; TEST CHECK THAT THE SBI DRIVES THE PA ADDRESS
;3855 ; LINES ABOVE MEMORY PROPERLY.  THE TEST ASSURES THAT
;3856 ; THE ADDRESS BITS ARE NOT STUCK AND ARE INDEPENDENT.
;3857 ;
;3858 ; SEQUENCE OF EVENTS
;3859 ;
;3860 ; 1)  INVALIDATE ALL OF CACHE
;3861 ; 2)  INVALIDATE LOCATION 0
;3862 ; 3)  SET STARTING ADDRESS OF MEMORY TO ADDRESS
;3863 ;  BIT UNDER TEST
;3864 ; 4)  READ AT TEST LOCATION
;3865 ;  CHECK FOR A MISS
;3866 ; 5)  READ LOCATION ZERO (0)
;3867 ;  SEE TIME OUT
;3868 ;  IF RESPONSE THEN ADDRESS BIT NOT
;3869 ;   DRIVEN BY SBI
;3870 ; 6)  READ TEST LOCATION
;3871 ;  SEE A HIT
;3872 ;  IF MISS INCORRECT ADDRESS DRIVEN ON THE PA BUS
;3873 ; 7)  INVALIDATE TEST LOCATION
;3874 ; 8)  REPEAT STEPS 2 - 7 UNTILL MAX ADDRESS (BIT #28)
;3875 ;  IS TESTED
;3876 ;
;3877 ;ERROR LOGOUT
;3878 ;
;3879 ; DATA: TEST ADDRESS (BIT UNDER TEST)
;3880 ;  GROUP UNDER TEST (0=GROUP0,1=GROUP1)
;3881 ;  MAXIMUM ADDRESS OF MEMORY
;3882 ;  UNDEFINED
;3883 ;  UNDEFINED
;3884 ;  UNDEFINED
;3885 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
;3886 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
;3887 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
;3888 ;  INTERRUPT FLAG:
;3889 ;   0=>NO INTERRUPT OCCURRED
;3890 ;   1=>AN INTERRUPT OCCURRED
;3891 ;
;3892 ;-
;3893 ;*****************************************************************************
;3894 ;
U 1112, 0000.003D.0180.0800.0000.11B7  ;3895 =0 SUBTEST
U 1113, 0000.003C.0180.0800.0000.1114  ;3896  NOP
U 1114, 0000.003D.0180.0800.0000.11DB  ;3897 =0 CALL[INIT]
U 1115, 0000.003C.2180.0800.0084.72E1  ;3898  SC_K[.14]   ; Point to the First Meg
U 12E1, 0003.001C.0180.0AC8.0000.12E2  ;3899  R[9]_NOT.MASK   ; SAVE STARTING ADDRESS IN R[9]
U 12E2, 0003.003C.0180.09E8.0000.12E3  ;3900  RC[0D]_0   ;SET GROUP FLAG
U 12E3, 0003.003C.0180.09F0.0000.1116  ;3901 SXX110: RC[0E]_0   ;INVALIDATE ALL OF CACHE
U 1116, 0000.003D.0180.0800.0000.11D0  ;3902 =0 CALL[INVALIDATECACHE]  ; INVALIDATE THE GROUP UNDER TEST
```

```
U 1117. 0800.003C.0180.0A48.0000.12E4  ;3903 SXX130: D_R[9]     ;SET INITIAL TEST LOCATION
U 12E4. 0001.003C.0180.09F0.0000.1118  ;3904  RC[0E]_D
U 1118. 0000.003D.0180.0800.0000.119C  ;3905 =0 ERLOOP
U 1119. 0000.003C.0180.0800.0000.111A  ;3906 SXX140: NOP
U 111A. 0000.003D.0180.0800.0000.11DB  ;3907 =0 CALL[INIT]
U 111B. 0818.0038.4580.0800.0000.12E5  ;3908  D_K[.8000]    ;SET GROUP UNDER TEST
U 12E5. 0010.0038.0180.0968.0010.12E6  ;3909  ALU_RC[0D],CLK.UBCC
U 12E6. 0600.013C.0180.0800.0000.111C  ;3910  Z?,D_D.RIGHT    ;GROUP 0 UNDER TEST
U 111C. 0600.003C.0180.0800.0000.111D  ;3911 =0 D_D.RIGHT    ;GROUP 1 UNDER TEST
U 111D. 0000.003C.7580.3C00.0000.12E7  ;3912  ID[MAINT]_D    ;WRITE THE I.D. REGISTER
U 12E7. 0810.0038.0180.0970.0000.12E8  ;3913  D_RC[0E]    ;SET STARTING ADDRESS TO CONTROLLER
U 12E8. 0600.003C.6180.0800.0084.72E9  ;3914  D_D.RIGHT,SC_K[.F]  ;SHIFT TO STARTING ADDRESS POSITION
U 12E9. 0000.003C.0580.0800.0084.B2EA  ;3915  SC_SC-1
U 12EA. 0801.001C.0180.0800.0000.12EB  ;3916  D_D.ORNOT.MASK    ;SET INHIBIT ECC
U 12EB. 0010.0038.0180.0950.0200.12EC  ;3917  VA_RC[0A]    ;SET ADDRESS CONFIG REG B
U 12EC. 0000.003C.0180.0803.0000.12ED  ;3918  VA_VA+4      ; ...
U 12ED. 0000.003C.0180.A800.0000.111E  ;3919  MCT/WRITE.P    ;WRITE CONFIG REG B
        ;3920 =0 D_K[.6].    ; Wait for the write to colmplete
U 111E. 0818.0039.D580.0800.0000.11D9  ;3921   CALL[WAIT.LOOP]  ;
U 111F. 0018.0038.1980.0800.0200.12EE  ;3922  VA_K[ZERO]    ;INVALIDATE LOCATION ZERO
U 12EE. 0000.003C.0180.9000.0000.12EF  ;3923  MCT/INVALIDATE
U 12EF. 0010.0038.0180.0970.0200.12F0  ;3924  VA_RC[0E]    ;INVALIDATE TEST LOCATION
U 12F0. 0000.003C.0180.9000.0000.12F1  ;3925  MCT/INVALIDATE
U 12F1. 0010.0038.0180.0970.0200.12F2  ;3926  VA_RC[0E]    ;READ TEST LOCATION, SEE MISS
U 12F2. 0000.003C.0180.C800.0000.1080  ;3927  MCT/READ.P
U 1080. 0000.003D.0180.0800.0000.1030  ;3928 =00 CALL[NOINTR]
U 1081. 0000.003D.0180.0800.0000.119D  ;3929  ERROR1     ;UNEXPECTED INTERRUPT!!!!
U 1082. 0000.003C.75F0.2C00.0000.1083  ;3930  Q_ID[MAINT]
U 1083. 000D.2034.0180.0A00.0010.12F3  ;3931  ALU_Q[AND]R[0],CLK.UBCC  ;SEE A MISS
U 12F3. 0000.013C.0180.0800.0000.1120  ;3932  Z?
U 1120. 0000.003D.0180.0800.0000.119D  ;3933 =0 ERROR1     ;GOT A HIT, DIDN'T INVALIDATE
        ;3934       ;  TEST LOCATION PROPERLY
U 1121. 0F18.0038.1980.0800.0200.12F4  ;3935  VA_K[ZERO],D_0    ;READ LOCATION 0,SEE TIMEOUT
U 12F4. 0000.003C.8980.0800.0084.72F5  ;3936  SC_K[.D]    ;
U 12F5. 0801.001C.0180.0AF8.0000.12F6  ;3937  R[0F]_ALU,D_D.ORNOT.MASK ;CATCH TIMEOUT
U 12F6. 0000.003C.0180.C800.0000.12F7  ;3938  MCT/READ.P
U 12F7. 0000.003C.0180.0A78.0010.12F8  ;3939  ALU_R[0F],CLK.UBCC  ;CHECK TIME OUT
U 12F8. 0003.013C.0180.0AF8.0000.1122  ;3940  Z?,R[0F]_0
U 1122. 0000.003D.0180.0800.0000.119D  ;3941 =0 ERROR1     ;NO TIMEOUT, SBI DIDN'T
        ;3942       ;  DRIVE THE TEST ADDRESS BIT TO
        ;3943       ;  THE CACHE
U 1123. 0010.0038.0180.0970.0200.12F9  ;3944  VA_RC[0E]    ;READ THE TEST LOCATION SEE, HIT
U 12F9. 0000.003C.0180.C800.0000.12FA  ;3945  MCT/READ.P
U 12FA. 0000.003C.75F0.2C00.0000.12FB  ;3946  Q_ID[MAINT]
U 12FB. 000D.2034.0180.0A00.0010.12FC  ;3947  ALU_Q[AND]R[0],CLK.UBCC  ;CHECK FOR HIT
U 12FC. 0000.013C.0180.0800.0000.1124  ;3948  Z?
U 1124. 0000.003C.0180.0800.0000.12FD  ;3949 =0 J/SXX145
U 1125. 0000.003D.0180.0800.0000.119D  ;3950  ERROR1     ;MISS AT TEST LOCATION,SBI
        ;3951       ;  DIDN'T DRIVE THE CORRECT
        ;3952       ;  ADDRESS TEST BIT TO CACHE
U 12FD. 0010.0038.0180.0970.0200.12FE  ;3953 SXX145: VA_RC[0E]    ;INVALIDATE THE TEST LOCATION
U 12FE. 0000.003C.0180.9000.0000.12FF  ;3954  MCT/INVALIDATE
U 12FF. 0810.0038.ED80.0970.0084.7300  ;3955  SC_K[.1B],D_RC[0E]  ;SEE IF BIT #28 IS SET IN
U 1300. 0000.003C.0580.0800.0084.9301  ;3956  SC_SC+K[.1]    ; TEST ADDRESS
U 1301. 0001.0024.0180.0800.0010.1302  ;3957  ALU_D.ANDNOT.MASK,CLK.UBCC
```

```
U 1302. 0500.013C.0180.0800.0000.1126  ;3958  Z?,D_D.LEFT
U 1126. 0000.003C.0180.0800.0000.1303  ;3959 =0 J/SXX150   ;BIT #28 SET,CHECK GROUP BEING
        ;3960        ; TEST
U 1127. 0001.003C.0180.09F0.0000.1119  ;3961  RC[0E]_D,J/SXX140  ;
U 1303. 0010.0038.0180.0968.0010.1304  ;3962 SXX150: ALU_RC[0D],CLK.UBCC  ;CHECK GROUP FLAG
U 1304. 0000.013C.0180.0800.0000.1128  ;3963  Z?
U 1128. 0000.003C.0180.0800.0000.1305  ;3964 =0 J/SXX190   ;IT'S SET END TEST
U 1129. 0018.0038.0580.09E8.0000.12E3  ;3965  RC[0D]_K[.1],J/SXX110  ;GO REPEAT TEST
U 1305. 0000.003C.0180.0800.0000.112A  ;3966 SXX190: NOP    ;END TEST
        ;3967 ;
        ;3968 ;=============================================================================
        ;3969 ;*
        ;3970 ; SUBTEST 3 INHIBITING OF SBI CYCLES ON CACHE PARITY ERRORS AND MISS
        ;3971 ;
        ;3972 ; THIS TEST CHECK THAT NO SBI CYCLES ARE STARTED(I.E. NO STALL)
        ;3973 ; IF A MISS OCCURS WITH A CACHE PARITY ERROR. THE SILO
        ;3974 ; IS USED TO CHECK THAT NO OPERATIONS OCCURE ON THE SBI.
        ;3975 ;
        ;3976 ; ERROR LOGOUT
        ;3977 ;
        ;3978 ; IF NONE, DIDN'T GET CACHE PARITY ERROR (U-TRAP)
        ;3979 ;
        ;3980 ; DATA: EXPECTED DATA
        ;3981 ;  DATA READ FROM SILO
        ;3982 ;  SILO COUNT( 0 = 1ST LOCATION AFTER LOCK)
        ;3983 ;  GROUP FLAG: 0=GROUP 0,1=GROUP 1
        ;3984 ;  UNDEFINED
        ;3985 ;  UNDEFINED
        ;3986 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
        ;3987 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
        ;3988 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
        ;3989 ;  INTERRUPT FLAG:
        ;3990 ;   0=>NO INTERRUPT OCCURRED
        ;3991 ;   1=>AN INTERRUPT OCCURRED
        ;3992 ;
        ;3993 ;-
        ;3994 ;=============================================================================
        ;3995 ;
        ;3996 ;
U 112A. 0000.003D.0180.0800.0000.11B7  ;3997 =0 SUBTEST
U 112B. 0003.003C.0180.09D8.0000.112C  ;3998  RC[0B]_0   ;SET GROUP FLAG
U 112C. 0000.003D.0180.0800.0000.119C  ;3999 =0 ERLOOP
U 112D. 0000.003C.0180.0800.0000.112E  ;4000 XX1010: NOP
U 112E. 0000.003D.0180.0800.0000.11DB  ;4001 =0 CALL[INIT]
U 112F. 0010.0038.0180.0950.0200.1306  ;4002  VA_RC[0A]   ; SET THE MEMORY STARTING ADDRESS
        ;4003  D_K[.8000],    ; TO ZERO
U 1306. 0818.0038.4580.0803.0000.1307  ;4004   VA_VA+4
U 1307. 0600.003C.0180.0800.0000.1308  ;4005  D_D.RIGHT
U 1308. 0000.003C.0180.A800.0000.1309  ;4006  CACHE.P_D[LONG]
U 1309. 0018.0038.1980.0800.0200.1130  ;4007  VA_K[ZERO]
U 1130. 0000.003D.1980.0800.0084.71BB  ;4008 =0 SETRCF[ZERO]
U 1131. 0818.0038.4580.0800.0000.130A  ;4009  D_K[.8000]
U 130A. 0600.003C.0180.0800.0000.130B  ;4010  D_D.RIGHT
U 130B. 0000.003C.7580.3C00.0000.130C  ;4011  ID[MAINT]_D   ;INVALIDAT GROUP 0
U 130C. 0818.0038.0580.0800.0000.130D  ;4012  D_K[.1]    ;SET DATA FOR INVALIDATE OPERATION
```

```
U 130D. 0000.003C.0180.9000.0000.130E  ;4013  MCT/INVALIDATE   ;SYNC:INVALIDATE GROUP 0
U 130E. 0818.0038.4580.0800.0000.130F  ;4014  D_K[.8000]
U 130F. 0200.003C.0180.0800.0000.1310  ;4015  D_D.RIGHT2   ;SET GROUP 1 FOR INVALIDATION
U 1310. 0000.003C.7580.3C00.0000.1311  ;4016  ID[MAINT]_D
U 1311. 0818.0038.0580.0800.0000.1312  ;4017  D_K[.1]    ;SET DATA FOR GROUP 1
U 1312. 0000.003C.0180.9000.0000.1313  ;4018  MCT/INVALIDATE   ;SYNC:INVALIDATE GROUP 1
U 1313. 0010.0038.0180.0958.0010.1314  ;4019  ALU_RC[0B],CLK.UBCC  ;WHICH GROUP TO TEST ??
U 1314. 0000.013C.0180.0800.0000.1132  ;4020  Z?
U 1132. 0000.003C.0180.0800.0000.1319  ;4021 =0 J/XX1020   ;GO SET FOR GROUP 1
U 1133. 0818.0038.61F8.0800.0000.1315  ;4022  D_K[.F],Q_0   ;SET GROUP 0, BYTE 0 PARITY ERROR
U 1315. 0000.003C.6580.0800.0084.7316  ;4023  SC_K[.10]
U 1316. 0000.003C.0580.0800.0084.9317  ;4024  SC_SC+K[.1]
U 1317. 0D00.003C.0180.0800.0000.1318  ;4025  D_DAL.SC
U 1318. 0000.003C.7580.3C00.0000.131F  ;4026  ID[MAINT]_D,J/XX1030  ;SET THE MAINTENANCE REGISTER
U 1319. 0818.0038.0DF8.0800.0000.131A  ;4027 XX1020: D_K[.3],Q_0   ;SET GROUP 1,BYTE 0 PARITY ERROR
U 131A. 0819.0030.0180.0800.0000.131B  ;4028  D_D.OR.K[.8]
U 131B. 0000.003C.6580.0800.0084.731C  ;4029  SC_K[.10]
U 131C. 0000.003C.0580.0800.0084.931D  ;4030  SC_SC+K[.1]
U 131D. 0D00.003C.0180.0800.0000.131E  ;4031  D_DAL.SC
U 131E. 0000.003C.7580.3C00.0000.131F  ;4032  ID[MAINT]_D   ;SET FORCE PARITY ERROR
U 131F. 0F00.003C.0180.0800.0084.7320  ;4033 XX1030: D_0,SC_K[.8]   ;SET TO CATCH CACHE PARITY U-TRAPS
U 1320. 0801.001C.0180.0AF8.0000.1321  ;4034  R[0F]_ALU,D_D.ORNOT.MASK
U 1321. 0818.0038.85F8.0800.0000.1322  ;4035  D_K[.C],Q_0   ;ENABLE SILO
U 1322. 0000.003C.ED80.0800.0084.7323  ;4036  SC_K[.1B]
U 1323. 0D00.003C.0180.0800.0000.1324  ;4037  D_DAL.SC
U 1324. 0000.003C.7180.3C00.0000.1325  ;4038  ID[COMP]_D   ;SYNC:ENABLE SILO
U 1325. 0000.003C.0180.C800.0000.1326  ;4039  MCT/READ.P   ;SYNC:READ MISS WITH PARITY(CACHE) ERROR
U 1326. 0000.003C.0180.0A78.0010.1327  ;4040  ALU_R[0F],CLK.UBCC
U 1327. 0003.013C.0180.0AF8.0000.1134  ;4041  Z?,R[0F]_0
U 1134. 0000.003D.0180.0800.0000.119D  ;4042 =0 ERROR1    ;NO CACHE PARITY(DATA) U-TRAP  =-
U 1135. 0000.003C.0180.0800.0000.1136  ;4043  NOP
          ;4044 =0 D_K[.10],   ;
U 1136. 0818.0039.6580.0800.0000.11D9  ;4045   CALL[WAIT.LOOP]
U 1137. 0000.003C.0180.0800.0000.1138  ;4046  NOP
          ;4047   ;=============================================================]=
          ;4048   ; SILO SHOULD HAVE FILLED, READ SEE ALL ZERO CONTENT
          ;4049   ;=============================================================
U 1138. 0003.003D.F580.09E0.0084.71BB  ;4050 =0 RC[0C]_0,SETRCF[.A]
U 1139. 0003.003C.0180.09F0.0000.1328  ;4051  RC[0E]_0   ;SET EXPECTED DATA
U 1328. 0000.003C.61F0.2C00.0000.1329  ;4052 XX1040: Q_ID[SILO]   ;READ SILO
U 1329. 0001.203C.0180.09E8.0010.132A  ;4053  RC[0D]_Q,CLK.UBCC  ;TEST SILO CONTENT
U 132A. 0810.0138.0180.0960.0000.113A  ;4054  Z?,D_RC[0C]
U 113A. 0000.003D.0180.0800.0000.119D  ;4055 =0 ERROR1    ;SILO CONTAINED NON-ZERO DATA  =-
U 113B. 0019.0020.6180.0800.0010.132B  ;4056  ALU_D[XOR]K[.F],CLK.UBCC ;FINISHED LOOP ?
U 132B. 0819.0114.0580.09E0.0000.113C  ;4057  Z?,RC[0C]_ALU,D_D+K[.1]
U 113C. 0000.003C.0180.0800.0000.1328  ;4058 =0 J/XX1040
U 113D. 0010.0038.0180.0958.0010.132C  ;4059  ALU_RC[0B],CLK.UBCC  ;CHECK GROUP FLAG
U 132C. 0018.0138.0580.09D8.0000.113E  ;4060  Z?,RC[0B]_K[.1]
U 113E. 0000.003C.0180.0800.0000.132D  ;4061 =0 J/XX1050   ;IT'S SET END SUBTEST
U 113F. 0000.003C.0180.0800.0000.112D  ;4062  J/XX1010   ;REPEAT LOOP WITH OTHER GROUP
U 132D. 0000.003C.0180.0800.0000.1140  ;4063 XX1050: NOP
          ;4064 ;
          ;4065 ;=============================================================================
          ;4066 ;*
          ;4067 ; SUBTEST 4 CACHE PARITY ERRORS WITH FORCE MISS
```

```
                ;4068 ;
                ;4069 ; TIS TEST CHECKS TO SEE THAT SBI CYCLES ARE STARTED WHEN CACHE
                ;4070 ; PARITY ERROR EXSIST AND THE FORCE MISS BIT ARE SET. THE SILO
                ;4071 ; IS USED TO SEE THAT A SBI CYCLE(I.E. EXTENDED READ C/A)
                ;4072 ; IS STARTED.
                ;4073 ;
                ;4074 ; ERROR LOGOUT
                ;4075 ;
                ;4076 ; DATA:   GROUP FLAG:0 = GROUP 0, 1 = GROUP 1
                ;4077 ;  UNDEFINED
                ;4078 ;  UNDEFINED
                ;4079 ;  UNDEFINED
                ;4080 ;  UNDEFINED
                ;4081 ;  UNDEFINED
                ;4082 ;  ID[VECTOR] IFF UNEXPECTED INTERRUPT
                ;4083 ;  ID[SBI.ERR] IFF UNEXPECTED INTERRUPT
                ;4084 ;  ID[FAULT] IFF UNEXPECTED INTERRUPT
                ;4085 ;  INTERRUPT FLAG:
                ;4086 ;   0=>NO INTERRUPT OCCURRED
                ;4087 ;   1=>AN INTERRUPT OCCURRED
                ;4088 ;
                ;4089 ; SEE LISTING FOR FAILING EVENT
                ;4090 ;
                ;4091 ;-
                ;4092 ;;;
```

```
U 1341. 0F00.003C.0180.0800.0084.7342  ;4123 XX2030: D_0,SC_K[.8]   ;CATCH THE CACHE PARITY ERROR
U 1342. 0801.001C.0180.0AF8.0000.1343  ;4124  R[0F]_ALU,D_D.ORNOT.MASK
U 1343. 0818.0038.85F8.0800.0000.1344  ;4125  D_K[.C],Q_0   ;ENABLE THE SILO
U 1344. 0000.003C.ED80.0800.0084.7345  ;4126  SC_K[.1B]
U 1345. 0D00.003C.0180.0800.0000.1346  ;4127  D_DAL.SC
U 1346. 0000.003C.7180.3C00.0000.1347  ;4128  ID[COMP]_D   ;SYNC:ENABLE SILO
U 1347. 0000.003C.0180.C800.0000.1348  ;4129  MCT/READ.P   ;SYNC:READ WITH CACHE PARITY AND FORCE MISS
U 1348. 0000.003C.0180.0A78.0010.1349  ;4130  ALU_R[0F],CLK.UBCC  ;CHECK U-TRAP
U 1349. 0003.013C.0180.0AF8.0000.114A  ;4131  Z?,R[0F]_0
U 114A. 0000.003C.0180.0800.0000.134A  ;4132 =0 J/XX2040   ;NO CACHE PARITY U-TRAP
U 114B. 0000.003D.0180.0800.0000.119D  ;4133  ERROR1    ;GOT A CACHE PARITY U-TRAP  =-
          ;4134   ;=============================================================
          ;4135   ; READ SILO, SEE AN EXTENDED READ C/A
          ;4136   ;=============================================================
U 134A. 0003.003C.0180.09E0.0000.134B  ;4137 XX2040: RC[0C]_0   ;SET LOOP COUNTER
U 134B. 0000.003C.61F0.2C00.0000.134C  ;4138 XX2050: Q_ID[SILO]   ;READ SILO
U 134C. 0C1B.0000.65F8.0800.0082.134D  ;4139  D_Q,Q_0,SC_0-K[.10]  ;SHIFT SILO C/A TO BITS 8-0
U 134D. 0D00.003C.0180.0800.0000.134E  ;4140  D_DAL.SC
U 134E. 0818.0038.81E0.0800.0000.134F  ;4141  Q_D,D_K[.3FF]   ;MASK OF BIT 31-9
U 134F. 0600.003C.0180.0800.0000.1350  ;4142  D_D.RIGHT
U 1350. 001D.0034.1180.09E8.0084.7351  ;4143  RC[0D]_ALU,ALU_D[AND]Q,SC_K[.4] ;MASK AND SAVE
U 1351. 0000.003C.0580.0800.0084.9352  ;4144  SC_SC+K[.1]   ;GENERATE ^X E0
U 1352. 0818.0038.5D80.0800.0000.1353  ;4145  D_K[.7]
U 1353. 0D00.003C.0180.0800.0000.1354  ;4146  D_DAL.SC
U 1354. 0011.0020.0180.0968.0010.1355  ;4147  ALU_D[XOR]RC[0D],CLK.UBCC
U 1355. 0810.0138.0180.0960.0000.114C  ;4148  Z?,D_RC[0C]
U 114C. 0019.0020.6180.0800.0010.1356  ;4149 =0 J/XX2060,ALU_D[XOR]K[.F],CLK.UBCC ;NO MATCH,INCREMENT COUNTER
U 114D. 0010.0038.0180.0970.0010.1357  ;4150  J/XX2070,ALU_RC[0E],CLK.UBCC ;SILO MATCH END TEST, CHECK FLAG
U 1356. 0819.0114.0580.09E0.0000.114E  ;4151 XX2060: Z?,RC[0C]_ALU,D_D+K[.1]
U 114E. 0000.003C.0180.0800.0000.134B  ;4152 =0 J/XX2050   ;GO REPEAT SILO READ
U 114F. 0000.003D.0180.0800.0000.119D  ;4153  ERROR1    ;DIDN'T SEE AN EXT READ C/A IN SILO =-
U 1357. 0000.013C.0180.0800.0000.1150  ;4154 XX2070: Z?
U 1150. 0000.003C.0180.0800.0000.1358  ;4155 =0 J/T1CEND   ;GROUP FLAG SET, END SUBTEST
U 1151. 0018.0038.0580.09F0.0000.1143  ;4156  J/XX2010,RC[0E]_K[.1]  ;REPEAT LOOP FOR OTHER GROUP
U 1358. 0000.003C.0180.0800.0000.1152  ;4157 T1CEND: NOP    ; ALL DONE.
          ;4158
          ;4159
```

```
;4160   .PAGE "TEST 1FC ROM CHECKSUM TEST"
;4161  ;*************************************************************
;4162  ;+++
;4163  ;
;4164  ; Functional Description:
;4165  ; ---------------------
;4166  ;
;4167  ; This test  checks the  bootstrap ROMs on the M8213
;4168  ; or the M8376 boards (whichever are found depending
;4169  ; on the System configuration.
;4170  ; The following checks will be done:
;4171  ;
;4172  ; - The  first  location  will  be  read  and  it is
;4173  ;   expected to return the following data: x071159D4
;4174  ;   (this  is  valid  for  both  the MS780-C and the
;4175  ;   MS780-E ROMs)
;4176  ;
;4177  ; - Read a ROM location whose contents should be 0.
;4178  ;   This location is x20003800 for both the MS780-C
;4179  ;   and the MS780-E)
;4180  ;
;4181  ; - OR together the  contents of two locations whose
;4182  ;   result will be all ones.
;4183  ;   These locations are for the MS780-C:
;4184  ;
;4185  ;  20003010 which contains 51FFFFFF
;4186  ;    and
;4187  ;  2000331C which contains EFDE5212
;4188  ;
;4189  ;  (This data is  valid as of: 12-5-78 and it
;4190  ;   may change with future ECO's)
;4191  ;
;4192  ;   For the MS780-E the two locations are:
;4193  ;
;4194  ;  200030AC which contains A5A5A5A5
;4195  ;    and
;4196  ;  200030B0 which contains 5A5A5A5A
;4197  ;
;4198  ;  (This data is valid as of: 8-26-82 and it
;4199  ;   may also change with future ECO's)
;4200  ;
;4201  ; - Add the 32 bit CHECKSUM of the data in the ROMs.
;4202  ;   The expected checksum for the MS780-C ROMs is:
;4203  ;
;4204  ;  25CCD7E5 (As of 12-1-78 and it may
;4205  ;     change with future ECO's)
;4206  ;
;4207  ;   The expected checksum for the MS780-E ROMs is:
;4208  ;
;4209  ;  0E2734EA (As of 7-22-82 and it may
;4210  ;     also  change with future
;4211  ;     ECO's.   Note:  for  the
;4212  ;     MS780-E  this   data  is
;4213  ;     blasted in  the last ROM
;4214  ;     location)
```

```
;4215 ;
;4216 ; The following conditions  will result in an error
;4217 ; call:
;4218 ;
;4219 ; - No  memory controller  has been  found with its
;4220 ;   ROMs enabled.
;4221 ;
;4222 ; - Both MS780-C and  MS780-E memories found on the
;4223 ;   SBI, and the MS780-C has its ROMs enabled.
;4224 ;   (Note: If there  are both  type of  memories on
;4225 ;   SBI, i.e. MS780-C and MS780-E, ONLY the MS780-E
;4226 ;   should have its ROMs enabled.
;4227 ;
;4228 ; - Data  in the  first location does not match the
;4229 ;   expected value.
;4230 ;
;4231 ; - The OR result of  the  two locations  does  not
;4232 ;   equal xFFFFFFFF (see above).
;4233 ;
;4234 ; - The  location  that  should  read  zero returns
;4235 ;   other data than expected.
;4236 ;
;4237 ; - The added checksum does not equal the expected.
;4238 ;
;4239 ; - More than one memory controller  found with its
;4240 ;   ROMs enabled.
;4241 ;
;4242 ;
;4243 ;
;4244 ; Error Logout:
;4245 ; ------------
;4246 ;
;4247 ; See individual error reports.
;4248 ;
;4249 ;
;4250 ;=============================================================
;4251 ;
;4252 ; Register Map:
;4253 ; ------------
;4254 ;
;4255 ;     RA,RB     Description:
;4256 ;     -----     -----------
;4257 ;
;4258 ;
;4259 ; A Flag indicating the  type of memory chips on
;4260 ;  the arrays of the memory inder test.
;4261 ;
;4262 ;    00 = MS780-A   4K CHIPS
;4263 ;    01 = MS780-C  16K CHIPS
;4264 ;    10 = MS780-E  64K CHIPS
;4265 ;    11 = MS780-H 256K CHIPS
;4266 ;
;4267 ; B Flag indicating the type of the memory found
;4268 ;  at the current TR.
;4269 ;
```

```
            ;4270 ;    00 = Illegal Code (Not used)
            ;4271 ;    01 = MS780-E found
            ;4272 ;    10 = MS780-C found
            ;4273 ;    11 = No memory at this TR
            ;4274 ;
            ;4275 ;         C Flag indicating  what types of memories were
            ;4276 ;   found on the SBI:
            ;4277 ;
            ;4278 ;    00 = No memory controller found
            ;4279 ;    01 = Found MS780-E
            ;4280 ;    10 = Found MS780-C
            ;4281 ;    11 = BOTH MS780-C and MS780-E found
            ;4282 ;
            ;4283 ; D Flag  indicating  on which  type of  memory
            ;4284 ;   the ROMs were enabled.
            ;4285 ;
            ;4286 ;    00 = No ROMs found enabled
            ;4287 ;    01 = ROMs enabled on MS780-E
            ;4288 ;    10 = ROMs enabled on MS780-C
            ;4289 ;    11 = Not used
            ;4290 ;
            ;4291 ;
            ;4292 ;       RC       Description:
            ;4293 ;       --       -----------
            ;4294 ;
            ;4295 ; C RECEIVED DATA
            ;4296 ;
            ;4297 ; D EXPECTED DATA
            ;4298 ;
            ;4299 ; E TR of the Memory Controller with ROMs enabled
            ;4300 ;
            ;4301 ;--
            ;4302 ;*************************************************************
            ;4303 =0
U 1152,  0000,003D,0180,0800,0000,103A  ;4304 T.47: NEWTST     ; Signal beginning of a new test
U 1153,  0003,003C,0180,0AD0,0000,1359  ;4305  R[0A]_0     ; Clear Type of Memory Chip
U 1359,  0003,003C,0180,0AD8,0000,135A  ;4306  R[0B]_0     ; Clear "Current memory " flag
U 135A,  0003,003C,0180,0AE0,0000,135B  ;4307  R[0C]_0     ; Clear Flag indicating the types of
            ;4308       ; ...memory controllers found on the SBI
U 135B,  0003,003C,0180,0AE8,0000,135C  ;4309  R[0D]_0     ; Clear Flag indicating the type of
            ;4310       ; ...memory that was found with its ROMs
            ;4311       ; ...enabled
U 135C,  0818,0038,0580,0800,0000,135D  ;4312  D_K[.1]     ; Initialize TR pointer
U 135D,  0000,003C,9180,3C00,0000,1084  ;4313  ID[24]_D    ; ...
            ;4314 =00
            ;4315 T.47.LP0:
U 1084,  0000,003D,0180,0800,0000,11F5  ;4316  CALL[FNDMEM]    ; Is there a memory at this TR?
            ;4317       ; ...also set "Current memory" Flag
U 1085,  0000,003C,0180,0800,0000,1185  ;4318  J/NEXTTR    ; No Memory at this TR
U 1086,  0800,003C,0180,0A58,0000,135E  ;4319  D_R[0B]     ; Get Current Memory Type Flag
            ;4320 =
U 135E,  081C,2030,0180,0A60,0000,135F  ;4321  D_D.OR.R[0C]    ; Update Memory Found Flag
U 135F,  0001,003C,0180,0AE0,0000,1360  ;4322  R[0C]_D     ; Save in R[0C]
            ;4323  ALU_R[0B][XOR]K[.1],  ; Is the Current memory a M780-E?
U 1360,  0018,0020,0580,0A58,0010,1361  ;4324   CLK.UBCC   ; ...
```

```
U 1361, 0000,013C,0180,0800,0000,1156  ;4325  Z?
U 1156, 0000,003C,0180,0800,0000,1088  ;4326 =0 J/NOT.FOUND.E   ; Current Memory is not a MS780-E
          ;4327       ; OK. Found a MS780-E memory
          ;4328 FOUND.E:
          ;4329  ALU_R[0D][AND]K[.2],  ; Did we previously found ROMs
U 1157, 0018,0034,0980,0A68,0010,1362  ;4330   CLK.UBCC  ; ...enabled on a MS780-C/A?
U 1362, 0000,013C,91F0,2C00,0000,1158  ;4331  Z?,Q_ID[24]   ; Get TR level of current memory
```

```
            ;4332  .PAGE
            ;4333 ;
            ;4334 ;////////////////////////////////////////////////////////////////////////////////
            ;4335 ;
            ;4336
            ;4337 =0 ERROR2[.2],   ; ROMs were found enabled on a
U 1158, 0001,203D,0980,09E8,0084,7028  ;4338   RC[0D]_Q  ; ...MS780-C and a MS780-E was found
            ;4339       ; ...on the SBI
            ;4340 ;
            ;4341 ;////////////////////////////////////////////////////////////////////////////////
            ;4342 ;
            ;4343 ; ERROR LOGOUT:
            ;4344 ;
            ;4345 ; DATA: TR LEVEL OF MS780-C WITH ROMS ENABLED
            ;4346 ;  TR LEVEL OF CURRENTLY FOUND MS780-E
            ;4347 ;
            ;4348 ;////////////////////////////////////////////////////////////////////////////////
            ;4349 ;
U 1159, 0000,003C,0180,0800,0000,1088  ;4350  NOP
            ;4351 =00
            ;4352 NOT.FOUND.E:
U 1088  0000,003D,0180,0800,0000,1209  ;4353  CALL[FNDROM]   ; Are the ROMs enabled on this
            ;4354       ; ...controller?
U 1089, 0000,003C,0180,0800,0000,1185  ;4355  J/NEXTTR    ; ROMs are not enabled.
U 108A, 0001,003C,0180,09E0,0000,1363  ;4356  RC[0C]_D    ; Save data returned from the
            ;4357       ; ...first ROM location
            ;4358 =
            ;4359  D_R[0D],    ; Did we find other ROMs enabled
U 1363, 0800,003C,0180,0A68,0010,1364  ;4360    CLK.UBCC  ; ...before this one?
U 1364, 0000,013C,91F0,2C00,0000,115A  ;4361  Z?,Q_ID[24]    ; Get TR level of this memory
            ;4362       ; ...second set of ROMs enabled
```

```
            ;4363  .PAGE
            ;4364 ;
            ;4365 ;////////////////////////////////////////////////////////////////////////////////
            ;4366 ;
            ;4367
            ;4368 =0 RC[0D]_Q,    ; Error condition.  This is the
U 115A, 0001,203D,0980,09E8,0084,7028  ;4369  ERROR2[.2]    ; Found More than one Memory
            ;4370       ; ...controller with ROMs enabled
            ;4371 ;
            ;4372 ;////////////////////////////////////////////////////////////////////////////////
            ;4373 ;
            ;4374 ; ERROR LOGOUT:
            ;4375 ;
            ;4376 ; DATA: TR LEVEL OF FIRST MEMORY CONTROLLER WITH ROMS ENABLED
            ;4377 ;  TR LEVEL OF SECOND MEMORY CONTROLLER FOUND WITH ROMS ENABLED
            ;4378 ;
            ;4379 ;////////////////////////////////////////////////////////////////////////////////
            ;4380 ;
            ;4381 FIRST.ROM.FOUND:
            ;4382       ; OK. This is the first set of ROMs
            ;4383       ; ...found enabled
U 115B, 0800,003C,0180,0A58,0000,1365  ;4384  D_R[0B]     ; Set the Type of Memory found with
            ;4385       ; ...ROMs enabled
            ;4386  R[0D]_D,    ; Save the Type of memory found with
            ;4387       ; ...ROMs enabled
U 1365, 0001,003C,91F0,2EE8,0000,1366  ;4388   Q_ID[24]  ; Get TR level of this memory
            ;4389       ; ...controller
U 1366, 0001,203C,0180,09F0,0000,1367  ;4390  RC[0E]_Q    ; Save in RC[0E]
            ;4391  ALU_R[0B][AND]K[.2],  ; Is the current memory a MS780-C
U 1367, 0018,0034,0980,0A58,0010,1368  ;4392   CLK.UBCC  ; ...
U 1368, 0000,013C,0180,0800,0000,115C  ;4393  Z?
U 115C, 0000,003C,0180,0800,0000,1369  ;4394 =0 J/ROMS.ON.C    ; The current set of ROMs is found on
            ;4395       ; ...a MS780-C memory Controller
U 115D, 0000,003C,0180,0800,0000,1160  ;4396  J/CONTINUE.ROM.TEST  ; The current set of ROMs is NOT
            ;4397       ; ...on a MS780-C memory Controller
            ;4398 ROMS.ON.C:
            ;4399  ALU_R[0C][AND]K[.1],  ; Did we found a MS780-E at a previous
U 1369, 0018,0034,0580,0A60,0010,136A  ;4400   CLK.UBCC  ; ...TR level with ROMs not enabled?
U 136A, 0000,013C,0180,0800,0000,115E  ;4401  Z?
```

```
            ;4402  .PAGE
            ;4403 ;
            ;4404 ;////////////////////////////////////////////////////////////////////////////////
            ;4405 ;
U 115E, 0000,003D,0580,0800,0084,7024  ;4406 =0 ERROR1[1]   ; ROMs were found enabled on a
            ;4407       ; ...MS780-A/C and a MS780-E/H
            ;4408       ; ...exists on the SBI
U 115F, 0000,003C,0180,0800,0000,1160  ;4409  NOP
            ;4410 ;
            ;4411 ;////////////////////////////////////////////////////////////////////////////////
            ;4412 ;
            ;4413 ; ERROR LOGOUT:
            ;4414 ;
            ;4415 ; DATA: TR LEVEL OF MS780-A/C FOUND WITH ROMS ENABLED
            ;4416 ;
            ;4417 ;////////////////////////////////////////////////////////////////////////////////
            ;4418 ;
            ;4419 =0
            ;4420 CONTINUE.ROM.TEST:
U 1160, 0000,003D,0180,0800,0000,1218  ;4421  CALL[FRSLOC]   ; Generate the data expected from the
            ;4422       ; ...first location
U 1161, 0001,003C,0180,09E8,0000,136B  ;4423  RC[0D]_D   ; Save expected data in RC[0D]
            ;4424  ALU_D.XOR.RC[0C],  ; Does it equal the received?
U 136B, 0011,0020,0180,0960,0010,136C  ;4425   CLK.UBCC  ; ...
U 136C, 0000,013C,0180,0800,0000,1162  ;4426  Z?    ;
```

```
            ;4427  .PAGE
            ;4428 ;
            ;4429 ;////////////////////////////////////////////////////////////////////////////////
            ;4430 ;
U 1162, 0000,003D,0D80,0800,0084,7028  ;4431 =0 ERROR2[.3]   ; First location in ROM does not
            ;4432       ; ...equal the expected value
            ;4433
            ;4434 ;
            ;4435 ;////////////////////////////////////////////////////////////////////////////////
            ;4436 ;
            ;4437 ; ERROR LOGOUT:
            ;4438 ;
            ;4439 ; DATA: TR LEVEL OF MEMORY CONTROLLER WITH ROMS ENABLED
            ;4440 ;  EXPECTED DATA FROM THE FIRST LOCATION IN ROM
            ;4441 ;  RECEIVED DATA FROM FIRST ROM LOCATION
            ;4442 ;
            ;4443 ;////////////////////////////////////////////////////////////////////////////////
            ;4444 ;
U 1163, 0000,003C,0180,0800,0000,1090  ;4445  NOP
U 1090, 0000,003D,0180,0800,0000,1233  ;4446 =00 CALL[RDZERO]   ; Read zeroes from ROM location
            ;4447  ERROR1[.3],   ; Read zero test failed
U 1091, 0003,003D,0D80,09E8,0084,7024  ;4448   RC[0D]_0  ; ...set expected value (zero)
U 1092, 0000,003C,0180,0800,0000,136D  ;4449  NOP
            ;4450 =
            ;4451  ALU_R[0B][XOR]K[.2],  ; Is this memory a MS780-C?
U 136D, 0018,0020,0980,0A58,0010,136E  ;4452   CLK.UBCC  ; ...
U 136E, 001B,0100,0580,09E8,0000,1164  ;4453  Z?,RC[0D]_0-K[.1]  ; Set expected data
U 1164, 000D,003C,0180,0800,0000,1098  ;4454 =0 J/FOUND.MSE   ; No. This is a MS780-E
U 1165, 000D,003C,0180,0800,0000,1094  ;4455  NOP
U 1094, 0000,003D,0180,0800,0000,123B  ;4456 =00 CALL[MSC.ALLONE]  ; Read two locations on a MS780-C
            ;4457       ; ...that ored together read ones
```

```
            ;4458  .PAGE
            ;4459 ;
            ;4460 ;////////////////////////////////////////////////////////////////////////////////
            ;4461 ;
U 1095, 0000,003D,0D80,0800,0084,7024  ;4462  ERROR1[.3]   ; Wrong data returned from ORing
            ;4463       ; ...of two locations (i.e., did not
            ;4464       ; ...receive all ones as expected)
            ;4465 ;
            ;4466 ;////////////////////////////////////////////////////////////////////////////////
            ;4467 ;
            ;4468 ; ERROR LOGOUT:
            ;4469 ;
            ;4470 ; DATA: TR LEVEL OF MEMORY CONTROLLER WITH ROMS ENABLED
            ;4471 ;   NOTE: CURRENT MEMORY IS A MS78-A/C
            ;4472 ;  EXPECTED RESULT OF ORING THE TWO LOCATIONS
            ;4473 ;  RECEIVED RESULT OF OR
            ;4474 ;
            ;4475 ;////////////////////////////////////////////////////////////////////////////////
            ;4476 ;
U 1096, 0000,003C,0180,0800,0000,109C  ;4477  J/ALLONE.OK   ; OK. Received all ones from the
            ;4478       ; ...ORing of the two locations
            ;4479 =
            ;4480 =00
            ;4481 FOUND.MSE:
U 1098, 0000,003D,0180,0800,0000,124B  ;4482  CALL[MSE.ALLONE]  ; OR together two ROM locations on a
            ;4483       ; ...MS780-E
```

```
            ;4484  .PAGE
            ;4485 ;
            ;4486 ;////////////////////////////////////////////////////////////////////////////////
            ;4487 ;
U 1099, 0000,003D,0D80,0800,0084,7024  ;4488  ERROR1[.3]   ; Result of ORing two locations on
            ;4489      ; ...A MS780-E did NOT return all ones
            ;4490 ;
            ;4491 ;////////////////////////////////////////////////////////////////////////////////
            ;4492 ;
            ;4493 ; ERROR LOGOUT:
            ;4494 ;
            ;4495 ; DATA: TR LEVEL OF MEMORY CONTROLLER WITH ROMS ENABLED
            ;4496 ;   NOTE: CURRENT MEMORY IS A MS780-E/H
            ;4497 ;  EXPECTED DATA FROM ORING THE TWO ROM LOCATIONS
            ;4498 ;  RETURNED DATA FROM THE OR OF THE TWO LOCATIONS
            ;4499 ;
            ;4500 ;////////////////////////////////////////////////////////////////////////////////
            ;4501 ;
U 109A, 0000,003C,0180,0800,0000,109C  ;4502  NOP
            ;4503 =
            ;4504 =00
            ;4505 ALLONE.OK:
U 109C, 0000,003D,0180,0800,0000,120F  ;4506  CALL[ADDUP]   ; Add the contents of the ROM
U 109D, 0000,003C,0180,0800,0000,109E  ;4507  NOP
            ;4508  ALU_R[0B][XOR]K[.2],  ; Are the ROMs enabled on MS780-A/C?
U 109E, 0018,0020,0980,0A58,0010,136F  ;4509   CLK.UBCC  ; ...
            ;4510 =
U 136F, 0000,013C,0180,0800,0000,1166  ;4511  Z?
U 1166, 0000,003C,0180,0800,0000,116A  ;4512 =0 J/ROMS.ON.MSE   ; NO. ROMs are on MS780-E/H
U 1167, 0001,203C,0180,09E0,0000,1168  ;4513  RC[0C]_Q   ; Save the received checksum
U 1168, 0000,003D,0180,0800,0000,1225  ;4514 =0 CALL[MSC.CHECKSUM]  ; Go find the checksum on MS780-A/C
            ;4515  RC[0D]_D,   ; Save the expected checksum
U 1169, 0001,003C,0180,09E8,0000,1371  ;4516   J/CHECK.RCVD.CHKSM ; Go check the received check sum
            ;4517 =0
            ;4518 ROMS.ON.MSE:
U 116A, 0001,203C,0180,09E0,0000,116B  ;4519  RC[0C]_Q   ; Get recieved checksum
U 116B, 0003,003C,0180,09E8,0000,1370  ;4520  RC[0D]_0   ; Save the expected checksum
U 1370, 0F00,003C,0180,0800,0000,1371  ;4521  D_0    ; Load it in the D register
            ;4522
            ;4523 CHECK.RCVD.CHKSM:
            ;4524  ALU_D.XOR.RC[0C],  ; Is the received checksum equal to
U 1371, 0011,0020,0180,0960,0010,1372  ;4525   CLK.UBCC  ; ...the expected?
U 1372, 0000,013C,0180,0800,0000,116C  ;4526  Z?    ; Check condition codes
```

```
            ;4527  .PAGE
            ;4528 ;
            ;4529 ;////////////////////////////////////////////////////////////////////////////////
            ;4530 ;
U 116C. 0000.003D.0D80.0800.0084.7024  ;4531 =0 ERROR1[.3]   ; Received checksum does NOT equal
            ;4532       ; ...the expected
            ;4533 ;
            ;4534 ;////////////////////////////////////////////////////////////////////////////////
            ;4535 ;
            ;4536 ; ERROR LOGOUT:
            ;4537 ;
            ;4538 ; DATA: TR LEVEL OF MEMORY CONTROLLER WITH ROMS ENABLED
            ;4539 ;  EXPECTED CHECKSUM
            ;4540 ;  RECEIVED CHECKSUM
            ;4541 ;
            ;4542 ;////////////////////////////////////////////////////////////////////////////////
            ;4543 ;
            ;4544
U 116D. 0000.003C.0180.0800.0000.116E  ;4545  NOP
            ;4546 =0 SC_K[.3],    ; Decrement TypeOut flag
U 116E. 0000.003D.0D80.0800.0084.71B1  ;4547   CALL[DECREMENT.TO] ; ...
U 116F. 0800.003C.0180.0A50.0000.1373  ;4548  D_R[0A]    ; Get type of chip flag
U 1373. 0000.193C.0180.0800.0000.108C  ;4549  DI-0?    ; Find out the type of memory chips
U 108C. 0000.003C.0180.0800.0000.1170  ;4550 =1100 J/MS780A.4K    ; MS780-A 4K chips
U 108D. 0000.003C.0180.0800.0000.1172  ;4551  J/MS780C.16K    ; MS780-C 16K chips
U 108E. 0000.003C.0180.0800.0000.1174  ;4552  J/MS780E.64K    ; MS780-E 64K chips
U 108F. 0000.003C.0180.0800.0000.1176  ;4553  J/MS780F.256K    ; MS780-H 256K chips
            ;4554
            ;4555 =0
            ;4556 MS780A.4K:
U 1170. 0818.0039.0580.0800.0000.11A1  ;4557  TYPE_MSG_1    ; Print "MS780-A 4K" message
U 1171. 0000.003C.0180.0800.0000.1178  ;4558  J/PRINT.TR    ; Print the rest of the message
            ;4559 =0
            ;4560 MS780C.16K:
U 1172. 0818.0039.0980.0800.0000.11A1  ;4561  TYPE_MSG_2    ; Print "MS780-C 16K" message
U 1173. 0000.003C.0180.0800.0000.1178  ;4562  J/PRINT.TR    ; Print the rest of the message
            ;4563 =0
            ;4564 MS780E.64K:
U 1174. 0818.0039.0D80.0800.0000.11A1  ;4565  TYPE_MSG_3    ; Print "MS780-E 64K" message
U 1175. 0000.003C.0180.0800.0000.1178  ;4566  J/PRINT.TR    ; Print the rest of the message
            ;4567 =0
            ;4568 MS780F.256K:
U 1176. 0818.0039.1180.0800.0000.11A1  ;4569  TYPE_MSG_4    ; Print "MS780-H 256K" message
U 1177. 0000.003C.0180.0800.0000.1178  ;4570  J/PRINT.TR    ; Print the rest of the message
            ;4571 =0
            ;4572 PRINT.TR:
U 1178. 0838.0039.0B80.0800.0000.11A1  ;4573  TYPE_MSG_5    ; Print " CHIP AT TR " message
U 1179. 0810.0038.0180.0970.0000.117A  ;4574  D_RC[0E]    ; Print TR level now
U 117A. 0000.003D.0180.0800.0000.11A5  ;4575 =0 TYP_RC[0E]_CRLF    ; ...
            ;4576  ALU_R[0B][XOR]K[.1],  ; Are these ROMs on a MS780-E/H?
U 117B. 0018.0020.0580.0A58.0010.1374  ;4577   CLK.UBCC   ; ...
U 1374. 0000.013C.0180.0800.0000.117C  ;4578  Z?
U 117C. 0000.003C.0180.0800.0000.1182  ;4579 =0 J/M8213.ROM    ; ROMs are on MS780-A/C
U 117D. 0000.003C.0180.0800.0000.117E  ;4580  NOP
            ;4581 =0 SC_K[.2],
```

```
U 117E, 0000,003D,0980,0800,0084,71B1  ;4582   CALL[DECREMENT.T0] ; Prepeare to print 2 more
        ;4583       ; ...messages
U 117F, 0000,003C,0180,0800,0000,1180  ;4584  NOP
U 1180, 0818,0039,5D80,0800,0000,11A1  ;4585 =0 TYPE_MSG_7    ; Print "M8376"
U 1181, 0000,003C,0180,0800,0000,1184  ;4586  J/PRINT.ROMS.OK   ; Go finish the message
        ;4587 =0
        ;4588 M8213.ROM:
U 1182, 0818,0039,D580,0800,0000,11A1  ;4589  TYPE_MSG_6    ; Print "M8213"
U 1183, 0000,003C,0180,0800,0000,1184  ;4590  NOP
        ;4591 =0
        ;4592 PRINT.ROMS.OK:
U 1184, 0818,0039,0180,0800,0000,11A2  ;4593  TYP_MSG_8_CRLF    ; Print " ROMS OK" message
        ;4594       ; We are done with these ROMs.
        ;4595       ; ...look for more
        ;4596 NEXTTR:
U 1185, 0000,003C,91F0,2C00,0000,1375  ;4597  Q_ID[24]    ; Get current TR level
        ;4598  ALU_Q.0XT[BYTE].XOR.K[.F], ; Are we at TR xF yet?
U 1375, 001B,A020,6180,0800,0010,1376  ;4599   CLK.UBCC  ; ...
        ;4600  Z?,
U 1376, 0819,2114,0580,0800,0000,1186  ;4601   D_Q+K[.1]  ; Increment TR level
        ;4602 =0 J/T.47.LP0,
U 1186, 0000,003C,9180,3C00,0000,1084  ;4603   ID[24]_D  ; NOT done yet. Save TR level.
        ;4604  D_R[0D],    ; Did we find any ROMs?
U 1187, 0800,003C,0180,0A68,0010,1377  ;4605   CLK.UBCC  ; ...
U 1377, 0000,013C,0180,0800,0000,1188  ;4606  Z?
U 1188, 0000,003C,0180,0800,0000,137F  ;4607 =0 J/T.47.EXIT    ; OK. We did find the right ROM
        ;4608       ; ...configuration
        ;4609  ALU_R[0C],    ; Did we find any memories on the SBI?
U 1189, 0000,003C,0180,0A60,0010,1378  ;4610   CLK.UBCC  ; ...
U 1378, 0000,013C,0180,0800,0000,118A  ;4611  Z?
U 118A, 0000,003C,0180,0800,0000,118C  ;4612 =0 J/FOUND.MEMORY    ; Yes. We found a memory controller
U 118B, 0000,003C,0180,0800,0000,137F  ;4613  J/T.47.EXIT    ; No memory controllers found.
        ;4614       ; ...exit the test
        ;4615 =0
        ;4616 FOUND.MEMORY:
        ;4617  SC_K[.2],    ; Prepare to type 2 messages
U 118C, 0000,003D,0980,0800,0084,71B1  ;4618   CALL[DECREMENT.T0] ; ...
U 118D, 0000,003C,0180,0800,0000,118E  ;4619  NOP
U 118E, 0818,0039,D980,0800,0000,11A2  ;4620 =0 TYP_MSG_9_CRLF    ; Print:
        ;4621       ; ...    NO M8213 OR 8376 ROMS"
U 118F, 0000,003C,0180,0800,0000,1190  ;4622  NOP
U 1190, 0818,0039,F580,0800,0000,11A2  ;4623 =0 TYP_MSG_A_CRLF    ; Print:
        ;4624       ; ..."DEPOSIT CONTROLLER ADDRESS IN ..."
U 1191, 0003,003C,0180,0980,0000,1192  ;4625  RC[0]_0
U 1192, 0000,003D,0180,0800,0000,119C  ;4626 =0 ERLOOP     ; Set the loop on error
        ;4627  VA_RC[0],    ; Get address of controller passsed
        ;4628       ; ...Mic Mon by operator
U 1193, 0010,0038,8980,0900,0284,7379  ;4629   SC_K[.D]  ; Prepare to enable Time-Out traps
U 1379, 081B,001C,C180,0800,0000,137A  ;4630  D_NOT.K[.FFFF]    ; Clear ID Fault Reg
U 137A, 0F00,003C,6D80,3C00,0000,137B  ;4631  ID[FAULT]_D,D_0    ; ...
U 137B, 0000,003C,6D80,3C00,0000,137C  ;4632  ID[FAULT]_D    ; ...
        ;4633  D[LONG]_CACHE.P,  ; Read Controller CNFG Reg A
U 137C, 0003,001C,0180,CAF8,0000,137D  ;4634   R[0F]_NOT.MASK  ; Enable Time-Out traps
U 137D, 0818,0038,0D80,0800,0000,137E  ;4635  D_K[.3]     ; ...
        ;4636  ERROR1[.1],    ; Call an error indicating that NO
```

```
U 137E, 0001,003D,0580,09F0,0084,7024  ;4637   RC[0E]_D  ; ...ROMs were found
            ;4638 T.47.EXIT:
U 137F, 0000,003C,0180,0800,0000,1194  ;4639  NOP    ; Exit the test now
            ;4640
```

```
            ;4641  .PAGE "TEST 1FD RESERVED"
            ;4642 =0
U 1194. 0000.003D.0180.0800.0000.103A  ;4643 T1FD: NEWTST
U 1195. 0000.003C.0180.0800.0000.1196  ;4644  NOP
            ;4645
```

```
            ;4646  .PAGE "TEST 1FE RESERVED"
            ;4647 =0
U 1196. 0000,003D,0180,0800,0000,103A  ;4648 T1FE: NEWTST
U 1197. 0000,003C,0180,0800,0000,1380  ;4649  NOP
            ;4650
```

```
          ;4651  .PAGE "DUMMY NEWTST"
U 1380. 0000,003D,0180,0800,0000,103A  ;4652 DUM: NEWTST
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1914: 1881 1888= 1904= 1905= 1906= 1882 2323=
U 1008  3333= 3334= 3335= 3337= 1965= 1967= 1883 2327=
U 1010  3382= 3383= 3384= 3386= 3485= 3486= 3487= 1884
U 1018  1885 2704= 1886 2705= 1887 2706= 1907 2708=
U 1020  1978= 1979= 1908 2715= 2002= 2003= 1909 2717=
U 1028  2026= 2027= 1910 2719= 2268= 2269= 1911 2720=
U 1030  2457= 2458= 1912 2727= 2461= 2462= 1913 2729=
U 1038  2464= 2465= 1954 2731= 2469= 2470= 1955 2732=
U 1040  2485= 2486= 1956 3838= 2491= 2492= 1957 3839=
U 1048  2526= 2527= 1958 3840= 2541= 2542= 1984: 3841=
U 1050  3492= 3493= 3494= 1959 3498= 3499= 3500= 1960
U 1058  2544= 2545= 2009: 1961 2547= 2548= 2033: 1962
U 1060  3504= 3507= 3509= 3510= 3558= 3559= 3566= 3568=
U 1068  3107= 3108= 3109= 3110= 3111= 3112= 3113= 3114=
U 1070  3799= 3800= 3801= 3802= 3815= 3816= 3817= 3818=
U 1078  3825= 3826= 3827= 3828= 3623= 3624= 3625= 3626=
U 1080  3928= 3929= 3930= 3931= 4316= 4318= 4319= 1963
U 1088  4353= 4355= 4356= 1964 4550= 4551= 4552= 4553=
U 1090  4446= 4448= 4449= 1968 4456= 4462= 4477= 1969
U 1098  4482= 4488= 4502= 1970 4506= 4507= 4509= 1971
U 10A0  2558= 2559= 2560= 2562= 2563= 2564= 2565= 2566=
U 10A8  2567= 2568= 2569= 2570= 2571= 2572= 2573= 2574=
U 10B0  2686= 2688= 2694= 2695= 2744= 2746= 2752= 2753=
U 10B8  2772= 2773= 2841= 2842= 2852= 2853= 2881= 2882=
U 10C0  2894= 2895= 2942= 2944= 2967= 2968= 3014= 3015=
U 10C8  3086= 3087= 3095= 3096= 3097= 3099= 3102= 3103=
U 10D0  3144= 3145= 3146= 3147= 3154= 3155= 3161= 3163=
U 10D8  3166= 3167= 3187= 3188= 3190= 3191= 3288= 3289=
U 10E0  3418= 3419= 3422= 3423= 3434= 3435= 3438= 3439=
U 10E8  3708= 3709= 3710= 3712= 3723= 3725= 3778= 3779=
U 10F0  3780= 3781= 3783= 3784= 3785= 3786= 3787= 3788=
U 10F8  3791= 3792= 3804= 3806= 3811= 3812= 3820= 3823=
U 1100  1868: 1869: 1870: 1871: 1872: 1873: 1874: 1875:
U 1108  1876: 1972 3830= 3831= 1877: 1878: 1879: 1880:
U 1110  3844= 3845= 3895= 3896= 3897= 3898= 3902= 3903=
U 1118  3905= 3906= 3907= 3908= 3911= 3912= 3921= 3922=
U 1120  3933= 3935= 3941= 3944= 3949= 3950= 3959= 3961=
U 1128  3964= 3965= 3997= 3998= 3999= 4000= 4001= 4002=
U 1130  4008= 4009= 4021= 4022= 4042= 4043= 4045= 4046=
U 1138  4050= 4051= 4055= 4056= 4058= 4059= 4061= 4062=
U 1140  4094= 4095= 4096= 4097= 4098= 4099= 4100= 4101=
U 1148  4113= 4114= 4132= 4133= 4149= 4150= 4152= 4153=
U 1150  4155= 4156= 4304= 4305= 1973 2053: 4326= 4330=
U 1158  4338= 4350= 4369= 4384= 4394= 4396= 4406= 4409=
U 1160  4421= 4423= 4431= 4445= 4454= 4455= 4512= 4513=
U 1168  4514= 4516= 4519= 4520= 4531= 4545= 4547= 4548=
U 1170  4557= 4558= 4561= 4562= 4565= 4566= 4569= 4570=
U 1178  4573= 4574= 4575= 4577= 4579= 4580= 4582= 4584=
U 1180  4585= 4586= 4589= 4590= 4593= 4597= 4603= 4605=
U 1188  4607= 4610= 4612= 4613= 4618= 4619= 4620= 4622=
U 1190  4623= 4625= 4626= 4629= 4643= 4644= 4648= 4649=
U 1198  1974 1975 1976 1977 1981 1982 1983 2007
```

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0 2008 2095 2136 2138 2178 2220 2263 2264
U 11A8 2267 2272 2312 2314 2315 2317 2319 2321
U 11B0 2329 2367 2368 2370 2394 2395 2396 2403
U 11B8 2404 2411 2412 2418 2419 2420 2430 2431
U 11C0 2432 2433 2434 2459 2460 2466 2467 2468
U 11C8 2471 2472 2473 2474 2475 2476 2477 2478
U 11D0 2481 2482 2483 2484 2487 2488 2489 2490
U 11D8 2494 2521 2539 2557 2575 2576 2585 2586
U 11E0 2587 2595 2596 2597 2598 2606 2607 2608
U 11E8 2609 2617 2618 2619 2620 2629 2630 2631
U 11F0 2632 2639 2640 2641 2642 2684 2689 2690
U 11F8 2692 2693 2698 2699 2700 2701 2702 2710
U 1200 2711 2712 2713 2714 2722 2723 2724 2725
U 1208 2726 2742 2743 2748 2749 2750 2751 2762
U 210 2763 2764 2765 2767 2768 2769 2770 2771
U .218 2785 2786 2787 2788 2789 2790 2791 2792
U 1220 2793 2794 2795 2796 2797 2810 2811 2812
U 1228 2813 2814 2815 2816 2817 2818 2819 2820
U 1230 2821 2822 2823 2840 2843 2844 2845 2846
U 1238 2848 2849 2850 2875 2876 2877 2878 2879
U 1240 2880 2883 2884 2886 2887 2888 2889 2890
U 1248 2891 2892 2893 2940 2945 2947 2948 2950
U 1250 2952 2953 2954 2956 2957 2958 2960 2962
U 1258 2963 2965 2966 2991 3016 3017 3018 3019
U 1260 3046 3047 3048 3088 3089 3090 3091 3100
U 1268 3101 3105 3106 3119 3120 3122 3123 3124
U 1270 3126 3131 3132 3133 3135 3140 3142 3143
U 1278 3151 3152 3156 3157 3158 3159 3160 3168
U 1280 3169 3170 3172 3173 3174 3175 3176 3177
U 1288 3183 3185 3186 3189 3214 3215 3216 3217
U 1290 3246 3247 3249 3250 3251 3277 3278 3279
U 1298 3280 3281 3282 3283 3284 3286 3287 3414
U 12A0 3416 3417 3421 3425 3426 3427 3428 3430
U 12A8 3431 3432 2054: 3433 3436 3437 3483 3484
U 12B0 3511 3552 3553 3554 3555 3556 3570 3572
U 12B8 3617 3618 3619 3621 3622 3713 3714 3715
U 12C0 3716 3717 3718 3719 3720 3726 3782 3789
U 12C8 3790 3793 3794 3795 3796 3797 3798 3803
U 12D0 3807 3808 3809 3810 3813 3814 3819 3824
U 12D8 3829 3834 3835 3836 3837 3842 3843 3846
U 12E0 3847 3899 3900 3901 3904 3909 3910 3913
U 12E8 3914 3915 3916 3917 3918 3919 3923 3924
U 12F0 3925 3926 3927 3932 3936 3937 3938 3939
U 12F8 3940 3945 3946 3947 3948 3953 3954 3955
U 1300 3956 3957 3958 3962 3963 3966 4004 4005
U 1308 4006 4007 4010 4011 4012 4013 4014 4015
U 1310 4016 4017 4018 4019 4020 4023 4024 4025
U 1318 4026 4027 4028 4029 4030 4031 4032 4033
U 1320 4034 4035 4036 4037 4038 4039 4040 4041
U 1328 4052 4053 4054 4057 4060 4063 4102 4103
U 1330 4104 4105 4106 4107 4108 4109 4110 4111
U 1338 4112 4115 4116 4117 4118 4119 4120 4121
U 1340 4122 4123 4124 4125 4126 4127 4128 4129

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1920
ESKAR53.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:55     Page  12

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  4130 4131 4137 4138 4139 4140 4141 4142
U 1350  4143 4144 4145 4146 4147 4148 4151 4154
U 1358  4157 4306 4307 4309 4312 4313 4321 4322
U 1360  4324 4325 4331 4360 4361 4388 4390 4392
U 1368  4393 4400 4401 4425 4426 4452 4453 4511
U 1370  4521 4525 4526 4549 4578 4599 4601 4606
U 1378  4611 4630 4631 4632 4634 4635 4637 4639
U 1380  4652
U 1388 - 13EF Unused
U 13F0  2037: 2038: 2039: 2040: 2041: 2042: 2043: 2044:
U 13F8  2045: 2046: 2047: 2048: 2049: 2050: 2051: 2052:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR53	913
Used	913
Remaining	

Total microwords used in memory U: 913
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 1922
ESKAR53.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:55     Page  12


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR53.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR54.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 1923

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2173 Diagnostic Specific Subroutines
; 2174   Select Controller
; 2212   START TEST
; 2421   SELECT LOWER CONTROLLER
; 2469   SELECT UPPER CONTROLLER
; 2517   Set Diagnostic Mode Routine
; 2546   SBI Unjam Routine
; 2595   RESET SUBTEST NUMBER
; 2617   Initialize the SBI
; 2644   Disable Cache Routine
; 2665   Enable Cache
; 2687   CHECK UTRAP
; 2719   Set Trap Mask
; 2747   Find MS780-E Memory
; 2935   Generate IO Address
; 2960   Set Starting Address
; 2994   ENABLE NEXT MS780-E
; 3046   Set ECC Bits
; 3076   MS780-E/H CNFG REG CLEANUP
; 3147   Wait Loop Routine
; 3179   DW780 INITIALIZE ROUTINE
; 3240   DW780 WAIT INIT ROUTINE
; 3293   DW780 EXTNDED MASKED WRITE
; 3397   DW780 LONGWORD READ ROUTINE
; 3490 TEST 1FF EXTENDED MASK WRITES
; 3731 TEST 200 CRD/RDS ERRORS ON EXTENDED MASKED WRITE
; 4013 TEST 201 LONGWORD READ
; 4256 TEST 202 CRD/RDS ERRORS ON LONGWORD READS
; 4828 TEST 203 RESERVED
; 4833 TEST 204 RESEREVED
; 4838 DUMMY NEWTST

```
:1 .NOLIST
:1804 .LIST
:1805
```

```
        ;1806 .REGION/1000.13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR54
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
;1841  .PAGE
;1842
;1843  .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844  .TOC "   Micro Trap Handler"
;1845 ;++
;1846 ; FUNCTIONAL DESCRIPTION:
;1847 ;
;1848 ; This routine is responsible for 'catching' micro-traps
;1849 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850 ; contains the Trap-Expected Mask. If the specified bit is set in
;1851 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852 ; an error call is made.
;1853 ;
;1854 ; INPUT PARAMETERS:
;1855 ;
;1856 ; R[0F] - Expected Trap Mask
;1857 ;  Trap Code Function
;1858 ;  --------      --------
;1859 ;       0  System Init
;1860 ;       1  Data Unaligned
;1861 ;       2  Cross Page
;1862 ;       3  TB Modify
;1863 ;       4  TB Protection
;1864 ;       6  TB Miss
;1865 ;       7  TB Parity
;1866 ;       8  Cache Parity
;1867 ;       9  Reserved Floating Operand
;1868 ;       C  Read Data Substitute
;1869 ;       D  Time out
;1870 ;       F  Control Store Parity Error
;1871 ;      10  Odd Address
;1872 ;
;1873 ; OUTPUT PARAMETERS:
;1874 ;
;1875 ; R[0F] - Mask bit is cleared.
;1876 ;
;1877 ; DATA: Micro PC of Micro Trap
;1878 ;  Trap Code (See Above)
;1879 ;  Fault Status Register
;1880 ;  SBI Error Register
;1881 ;  Timeout Address Register
;1882 ;  Maintenance Register
;1883 ;  Cache Parity Register
;1884 ;  TB Error Register 1
;1885 ;  TB Error Register 0
;1886 ;--
U 1100,  0000,003C,0180,0800,0084,7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101,  0000,003C,0580,0800,0084,7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102,  0000,003C,0980,0800,0084,7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103,  0000,003C,0D80,0800,0084,7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104,  0000,003C,1180,0800,0084,7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105,  0000,003C,D580,0800,0084,7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106,  0000,003C,D980,0800,0084,7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107,  0000,003C,5D80,0800,0084,7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108,  0000,003C,0180,0800,0084,7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
U 110C, 0000,003C,8580,0800,0084,7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000,003C,8980,0800,0084,7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000,003C,6580,0800,0084,7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000,003C,6180,0800,0084,7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
        ;1900
U 1001, 0000,003C,81F0,2C00,0000,1026  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 1026, 0C00,003C,01E0,0800,0000,102E  ;1902  q_d,d_q   ; Setup to put it back on stack
U 102E, 0C00,003C,81E0,3C00,0000,1033  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 1033, 0000,003C,01C0,0A78,0000,103B  ;1904  q_r[0f]   ; Get the Expected Trap Mask
        ;1905  alu_q.andnot.mask,
U 103B, 0001,2024,0180,0800,0010,1047  ;1906   clk.ubcc ; Was trap expected?
U 1047, 0000,013C,0180,0800,0000,1002  ;1907  z?
        ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
        ;1909       alu_q.and.mask, ; ...
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1910       return0  ; ...
U 1003, 0018,0038,1D80,09E8,0000,1024  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 1024, 0000,003D,D980,0800,0084,7199  ;1912 =0 setrcf[.9]  ; Set output count
U 1025, 0000,003C,49F0,2C00,0000,104F  ;1913  q_id[tber0]  ; Output all the error registers
U 104F, 0001,203C,4DF0,2DB0,0000,105B  ;1914  rc[6]_q,q_id[tber1] ; ...
U 105B, 0001,203C,65F0,2DB8,0000,105F  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 105F, 0001,203C,6DF0,2DD8,0000,1063  ;1916  rc[0b]_q,q_id[fault] ; ...
U 1063, 0001,203C,75F0,2DE0,0000,1067  ;1917  rc[0c]_q,q_id[maint] ; ...
U 1067, 0001,203C,79F0,2DC8,0000,1109  ;1918  rc[9]_q,q_id[parity] ; ...
U 1109, 0001,203C,69F0,2DC0,0000,117E  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 117E, 0001,203C,81F0,2DD0,0000,1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
        ;1921 1000: ERROR1[.C],  ;
U 1000, 0001,203D,8580,09F0,0084,7058  ;1922   rc[0e]_q ; Report the error
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "    Newtest Routine"
                ;1926 ;++
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to     0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 117F, 0000,003C,65F8,0800,0084,7180  ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1180, 0818,0038,8D80,0800,0000,1181  ;1955  d_k[.1f]  ; D=1F
U 1181, 0D00,003C,0180,0800,0000,1182  ;1956  d_dal.sc  ; D=001F0000
U 1182, 0000,003C,3D80,3C00,0000,1183  ;1957  id[psl]_d  ; psl = 001F0000
U 1183, 0818,0038,0580,0800,0000,1184  ;1958  d_k[.1]  ; Clear the CES register
U 1184, 0D00,003C,0180,0800,0000,1185  ;1959  d_dal.sc  ; D = 00010000
U 1185, 0F00,003C,3180,3C00,0000,1186  ;1960  id[ces]_d,d_0  ; ces = 00010000
U 1186, 0000,003C,5D80,3C00,0000,1187  ;1961  id[acc.cs]_d  ; acc.cs = 0
U 1187, 0000,003C,3980,3C00,0000,1188  ;1962  id[sir]_d  ; sir = 0
U 1188, 0000,003C,2980,3C00,0000,1189  ;1963  id[clk.cs]_d  ; clk.cs = 0
U 1189, 0000,003C,4980,3C00,0000,102C  ;1964  id[tber0]_d  ; tber0 = 0
U 102C, 0000,003D,0180,0800,0000,11B5  ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1966  intrpt.strobe,  ; Strobe interrupts
U 102D, 0818,0038,C180,0800,4000,118A  ;1967      d_k[.ffff]  ; Setup to clear SBI error register and
U 118A, 0801,0028,6580,3C00,0000,118B  ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 118B, 0F00,003C,6D80,3C00,0000,118C  ;1969  id[fault]_d,d_0  ; ...
U 118C, 0000,003C,6D80,3C00,0000,118D  ;1970  id[fault]_d  ; fault = 0
U 118D, 0000,003C,7580,3C00,0000,118E  ;1971  id[maint]_d  ; MAINT = 0
U 118E, 0000,003C,6580,3C00,0000,118F  ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 118F, 0000,003C,7180,3C00,0000,1190  ;1973  id[comp]_d  ; comp reg = 0
U 1190, 0000,003C,E580,3C00,0000,1191  ;1974  ID[39]_d  ; Clear code for subroutine START.TEST
U 1191, 0001,003C,0180,09F0,0000,1192  ;1975  rc[0e]_d  ;
U 1192, 0001,003C,0180,0AF8,0000,1193  ;1976  r[0f]_d  ; Clear expected trap mask
U 1193, 0001,003C,0180,09F8,0000,104C  ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 104C, 0000,003D,0180,0800,0000,11BB  ;1978 =0 call[disable.cache] ; Disable the cache
U 104D, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            ;1980   .PAGE
            ;1981   .TOC "   Error Loop Routine"
            ;1982 ;++
            ;1983 ; FUNCTIONAL DESCRIPTION:
            ;1984 ;
            ;1985 ; This routine is called with the "ERLOOP" macro. The
            ;1986 ; routine calls the Micro Monitor to setup an error loop.
            ;1987 ;
            ;1988 ; CALLING CONSTRAINT:
            ;1989 ;
            ;1990 ; =0
            ;1991 ;
            ;1992 ; SIDE EFFECTS:
            ;1993 ;
            ;1994 ; D Reg - Destroyed
            ;1995 ;--
U 1194, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 1195, 0810,0038,0180,0978,0000,1196  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 1196, 0819,0034,4D80,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E, 0819,0030,1180,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;++
            ;2025 ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 1058. 0000.003D,0180.0800.0000.1199  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1059. 0000.003C,0180.0800.0000.1195  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "   Error 2 Routine"
            ;2045 ;++
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 1197, 0810,0038,0180,0978,0000,1198  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 1198, 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "   ERROR 2 WITH PARAMETER"
            ;2071 ;++
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 105C, 0000,003D,0180,0800,0000,1199  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 105D, 0000,003C,0180,0800,0000,1197  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
            ;2088  .PAGE
            ;2089  .TOC "   Micro-Monitor Call"
            ;2090 ;*
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;++
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0. 0000.003C.0180.0800.0000.13F1   ;2111 13F0: nop
U 13F1. 0000.003C.0180.0800.0000.13F2   ;2112 13F1: nop
U 13F2. 0000.003C.0180.0800.0000.13F3   ;2113 13F2: nop
U 13F3. 0000.003C.0180.0800.0000.13F4   ;2114 13F3: nop
U 13F4. 0000.003C.0180.0800.0000.13F5   ;2115 13F4: nop
U 13F5. 0000.003C.0180.0800.0000.13F6   ;2116 13F5: nop
U 13F6. 0000.003C.0180.0800.0000.13F7   ;2117 13F6: nop
U 13F7. 0000.003C.0180.0800.0000.13F8   ;2118 13F7: nop
U 13F8. 0000.003C.0180.0800.0000.13F9   ;2119 13F8: nop
U 13F9. 0000.003C.0180.0800.0000.13FA   ;2120 13F9: nop
U 13FA. 0000.003C.0180.0800.0000.13FB   ;2121 13FA: nop
U 13FB. 0000.003C.0180.0800.0000.13FC   ;2122 13FB: nop
U 13FC. 0000.003C.0180.0800.0000.13FD   ;2123 13FC: nop
U 13FD. 0000.003C.0180.0800.0000.13FE   ;2124 13FD: nop
U 13FE. 0000.003C.0180.0800.0000.13FF   ;2125 13FE: nop
U 13FF. 0000.003C.0180.0800.0000.12AA   ;2126 13FF: nop
U 12AA. 0000.003C.0180.0800.0000.1199   ;2127 12AA: nop ; This location is permanently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;++
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 1199, 0010.0038,01C0,0978,0000,119A  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 119A, 0019.2034,4DC0,0800,0000,119B  ;2150  q_q.and.k[.ff00] ; ...
U 119B, 0019.2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "    Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 119C, 0810,0038,4D80,0978,0000,119D  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 119D, 0019,4016,4D80,09F8,0000,0001  ;2171       word,return1 ; (Subtest number is negative)
```

```
            ;2172  .PAGE
            ;2173  .TOC " Diagnostic Specific Subroutines"
            ;2174  .TOC "   Select Controller"
            ;2175 ;++
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine is used to put the memory system into the
            ;2179 ; specified configuration with respect to controller select.
            ;2180 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2181 ; a RETURN2 is made.
            ;2182 ;
            ;2183 ; CALLING CONSTRAINT:
            ;2184 ;
            ;2185 ; =00
            ;2186 ;
            ;2187 ; INPUTS:
            ;2188 ;
            ;2189 ; d - Contains value to write to bits<2:0> of Register A
            ;2190 ; rc[0e] - Address of Register A
            ;2191 ;
            ;2192 ; SIDE EFFECTS:
            ;2193 ;
            ;2194 ; sc,d,va are destroyed
            ;2195 ;--
            ;2196 SELECT.CNTRLR:
U 119E, 0010,0038,0180,0970,0284,719F  ;2197  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 119F, 0801,001C,0180,0800,0000,11A0  ;2198  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11A0, 0000,003C,0180,A800,0000,11A1  ;2199  cache.p_d[long]  ; Write the configuration Register
U 11A1, 0818,0038,5D80,0800,0000,11A2  ;2200  d_k[.7]  ; Get Misconfiguration bits
U 11A2, 0000,003C,61F8,0800,0084,71A3  ;2201  sc_k[.F],q_0  ; ...
U 11A3, 0D00,003C,0180,0800,0000,11A4  ;2202  d_dal.sc  ; ... D = x38000
U 11A4, 0000,003C,01E0,0800,0000,11A5  ;2203  q_d  ; Save mask
U 11A5, 0000,003C,0180,C800,0000,11A6  ;2204  d[long]_cache.p  ; Read CNFG A Reg and
            ;2205  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11A6, 001D,2034,0180,0800,0010,11A7  ;2206  clk.ubcc ; ...
U 11A7, 0000,013C,0180,0800,0000,106C  ;2207  z?  ; Branch if no
U 106C, 0000,003E,0180,0800,0000,0001  ;2208 =0 RETURN1  ; Bad configuration
U 106D, 0000,003E,0180,0800,0000,0002  ;2209  RETURN2  ; Configuration ok
            ;2210
```

```
;2211   .PAGE
;2212   .TOC "   START TEST"
;2213 ;*************************************************************
;2214 ;**
;2215 ;
;2216 ; Functional Description:
;2217 ; ----------------------
;2218 ;
;2219 ;         This subroutine will be called by all tests that  check  the
;2220 ;    controllers,  or  those  tests  that  have  to switch between the
;2221 ;    controllers when executing.   Its  main  functions  are:   select
;2222 ;    both  controllers  on the MS780-E/H SBI Interface and execute the
;2223 ;    test that called it, call out an error  if  neither  one  of  the
;2224 ;    controllers can be configured properly, if all the controllers on
;2225 ;    a MS780-E/H were configured and tested, it should select the next
;2226 ;    MS780-E/H on the SBI and repeat the above sequence.
;2227 ;    A test making use of this  subroutine  should  be  configured  as
;2228 ;    follows:
;2229 ;
;2230 ;
;2231 ;       Begin TEST.XX
;2232 ;        :
;2233 ;        :   Call NEWTST
;2234 ;        :   Call FIND.MS780E
;2235 ;        :   IF No MS780-E's on the SBI
;2236 ;        :     THEN
;2237 ;        :         Skip to End of TEST.XX
;2238 ;        :     ELSE
;2239 ;        :
;2240 ;        : RESTART.TEST.XX:
;2241 ;        :
;2242 ;        :   Call START TEST
;2243 ;        :   IF Return1 from START.TEST
;2244 ;        :     THEN
;2245 ;        :         Skip to End of TEST.XX
;2246 ;        :     ELSE.
;2247 ;        :
;2248 ;        ~
;2249 ;
;2250 ;        :
;2251 ;        :   Body of TEST.XX
;2252 ;        :
;2253 ;        ~
;2254 ;        ~
;2255 ;        :   Jump RESTART.TEST.XX
;2256 ;        :
;2257 ;       End TEST.XX
;2258 ;
;2259 ;
;2260 ;         This subroutine makes use of a pointer  in  ID  Register  39
;2261 ;    (Temporary  Register  9)  to  "remember"  at  which point control
;2262 ;    should be passed when the subroutine is called a second, third or
;2263 ;    fourth  time, depending on the number of MS780-E's on the SBI and
;2264 ;    the numbers of controllers on each.  (Note:  Temporary Register 9
;2265 ;    will  be  cleared  by  the  NEWTST subroutine.) The pointer in ID
```

```
;2266 ;    Register 39 can be interpreted as follows:
;2267 ;
;2268 ;     b00   -   Value that ID Reg 39 should hold when the subroutine is
;2269 ;               called the first time.  When this code is encountered
;2270 ;               this subroutine will try to select the lower controller.
;2271 ;               If the lower controller is misconfigured, the pointer in
;2272 ;               ID Reg 39  is changed to b01 (See below) and the subroutine
;2273 ;               is re-entered.
;2274 ;               If, however there is no misconfiguration, the value in
;2275 ;               ID Reg 39 is changed to b10 and a Return2 is made to the
;2276 ;               calling test.
;2277 ;
;2278 ;     b01   -   This value signifies that an attempt at configuring the
;2279 ;               lower controller failed, and the subroutine will attempt
;2280 ;               to select the upper controller.
;2281 ;               If the upper controller is also misconfigured execution
;2282 ;               stops with a call to ERROR1.
;2283 ;               If there is no misconfiguration on this controller, then
;2284 ;               the code in ID Reg 39 is changed to b11 and a Return2 is
;2285 ;               made to the calling test.
;2286 ;
;2287 ;     b10   -   This value signifies that the lower controller was selected
;2288 ;               previously and the test was executed once.  If the
;2289 ;               subroutine is entered with this code in ID Reg 39, ID Reg
;2290 ;               39 is loaded with b11 and an attempt is made to select the
;2291 ;               upper controller.
;2292 ;               If there is a misconfiguration on this controller, this
;2293 ;               subroutine is just re-entered.
;2294 ;               Otherwise, a Return2 will be made to the calling test.
;2295 ;
;2296 ;     b11   -   This code identifies the cases in which the test has
;2297 ;               been executed at least once (i.e., at least one of the
;2298 ;               controllers on the MS780-E unit under test could be
;2299 ;               configured properly).  When this code is detected the
;2300 ;               subroutine clears ID Reg 39 and makes a call to
;2301 ;               ENABLE.NEXT.MS780E.  If all MS780-E units have been
;2302 ;               tested then the subroutine executes a Return1.
;2303 ;               Otherwise the subroutine is re-entered.
;2304 ;
;2305 ;
;2306 ; Calling Constraint: =00
;2307 ; ------------------
;2308 ;
;2309 ;
;2310 ; Inputs:
;2311 ; ------
;2312 ;
;2313 ; ID Register 39 must be cleared by NEWTST
;2314 ;
;2315 ;
;2316 ; Outputs:
;2317 ; -------
;2318 ;
;2319 ; RC[0E] will have the I/O Base address of MS780-E/H to be tested
;2320 ; RC[0D] will be set up with the CNFG Register C/D of Controller
```

```
                ;2321 ;         to be tested
                ;2322 ;
                ;2323 ; Side Effects:
                ;2324 ; ------------
                ;2325 ;
                ;2326 ; Contents of the following regiters will be destroyed:
                ;2327 ;
                ;2328 ; D, Q
                ;2329 ;
                ;2330 ;--
                ;2331 ;*************************************************************
                ;2332 =0
                ;2333 START.TEST:
U 106E, 0000,003D,0180,0800,0000,11B6  ;2334  CALL[RESET.SUBTST]  ; Reset subtest number. (For those
U 106F, 0000,003C,0180,0800,0000,1070  ;2335  NOP     ; ...tests that have more than one
                ;2336 =0      ; ...subtest.)
U 1070, 0000,003D,0180,0800,0000,1194  ;2337  ERLOOP    ; Set up the error loop for the
                ;2338       ; ...eventuality that the MS780-E/H
                ;2339       ; ...currently under test has no
                ;2340       ; ...Controllers
                ;2341 RE.ENTRY:    ; Re entry point for this routine
U 1071, 0000,003C,E5F0,2C00,0000,11A8  ;2342  Q_ID[39]    ; Get the code
U 11A8, 0C00,003C,0180,0800,0000,11A9  ;2343  D_Q     ; Put it in the D Register
U 11A9, 0000,193C,0180,0800,0000,100C  ;2344  DI-0?     ; Branch on the code
U 100C, 0000,003C,0180,0800,0000,1004  ;2345 =1100 J/STRT.CASE00    ; Code is 00
U 100D, 0000,003C,0180,0800,0000,1008  ;2346  J/STRT.CASE01    ; Code is 01
U 100E, 0000,003C,0180,0800,0000,11AE  ;2347  J/STRT.CASE10    ; Code is 10
U 100F, 0000,003C,0180,0800,0000,1013  ;2348  J/STRT.CASE11    ; Code is 11
                ;2349 ;
                ;2350 ;
                ;2351 ; At this point the code in ID[39] was 00
                ;2352 ;
                ;2353 ;
                ;2354 =00
                ;2355 STRT.CASE00:
U 1004, 0000,003D,0180,0800,0000,1020  ;2356  CALL[SELECT.LOWER]  ; Try to select the lower Controller
                ;2357  J/STRT.LOW.MIS,    ; Lower Controller Misconfigured
U 1005, 0818,0038,0580,0800,0000,11AA  ;2358   D_K[.1]    ; Set up the code for re entry to this
                ;2359       ; ...routine ( 01 )
U 1006, 0818,0038,0980,0800,0000,1007  ;2360  D_K[.2]     ; Set up the code for a next call to
                ;2361       ; ...START.TEST indicating that the
                ;2362       ; ...Lower Controller was configured
                ;2363       ; ...( 10 )
                ;2364  ID[39]_D,    ; Save the code in ID[39]
U 1007, 0000,003E,E580,3C00,0000,0002  ;2365   RETURN2    ; Let the test know that the Lower
                ;2366       ; ...has been configured
                ;2367 STRT.LOW.MIS:
                ;2368  ID[39]_D,    ; Save code in ID[39] signifing that
U 11AA, 0000,003C,E580,3C00,0000,1071  ;2369   J/RE.ENTRY   ; ...the Lower Controller was Misconfig
                ;2370       ; ...and re enter this subroutine
                ;2371
                ;2372 ;
                ;2373 ;
                ;2374 ; At this point the code is 01
                ;2375 ;
```

```
            ;2376 ;
            ;2377 =00
            ;2378 STRT.CASE01:
U 1008, 0000,003D,0180,0800,0000,1028  ;2379  CALL[SELECT.UPPER]  ; Try to select the Upper Controller
            ;2380  J/STRT.UPR.MIS,   ; Upper Controller is Misconfigured
U 1009, 0F00,003C,0180,0800,0000,11AB  ;2381   D_0   ; Prepare to clear the code
U 100A, 0818,0038,0D80,0800,0000,100B  ;2382  D_K[.3]    ; Upper controller was configured
            ;2383       ; ...thus the code must be changed
            ;2384       ; ...accordingly ( 11 )
            ;2385  ID[39]_D,   ; Save the code
U 100B, 0000,003E,E580,3C00,0000,0002  ;2386   RETURN2   ; Let the test know that the Upper
            ;2387       ; ...Controller has been configured
            ;2388 STRT.UPR.MIS:
U 11AB, 0000,003C,E580,3C00,0000,11AC  ;2389  ID[39]_D   ; Save the Code ( 00 )
U 11AC, 0018,0038,0D80,09E8,0000,11AD  ;2390  RC[0D]_K[.3]   ; Set up a code for the Fail Chain
U 11AD, 0000,003D,0980,0800,0084,7058  ;2391  ERROR1[.2]   ; Flag the error.
            ;2392 ;
            ;2393 ;
            ;2394 ; At this point the code is 10
            ;2395 ;
            ;2396 ;
            ;2397 STRT.CASE10:
U 11AE, 0818,0038,0D80,0800,0000,1010  ;2398  D_K[.3]    ; Set the code to 11
            ;2399
            ;2400
            ;2401 =00 ID[39]_D,   ; Save the code
U 1010, 0000,003D,E580,3C00,0000,1028  ;2402   CALL[SELECT.UPPER] ; Try to select the upper Controller
U 1011, 0000,003C,0180,0800,0000,1071  ;2403  J/RE.ENTRY   ; Misconfigured Upper Controller
U 1012, 0000,003E,0180,0800,0000,0002  ;2404  RETURN2    ; Upper Controller configured
            ;2405       ; ...let the test know it
            ;2406 ;
            ;2407 ;
            ;2408 ; At this point the code is 11
            ;2409 ;
            ;2410 ;
            ;2411 STRT.CASE11:
U 1013, 0F00,003C,0180,0800,0000,1014  ;2412  D_0    ; Clear the code
            ;2413 =00 ID[39]_D,   ; Save the code
U 1014, 0000,003D,E580,3C00,0000,1215  ;2414   CALL[ENABLE.NEXT.MS780E]; See if there are more MS780-E/Hs
            ;2415       ; ...on the SBI
U 1015, 0000,003C,0180,0800,0000,1071  ;2416  J/RE.ENTRY   ; Found another MS780-E/H.   Go and
            ;2417       ; ...attempt to configure the cntrllrs
U 1016, 0000,003E,0180,0800,0000,0001  ;2418  RETURN1    ; No more MS780-E/Hs found
U 1017, 0000,003C,0180,0800,0000,1020  ;2419  NOP
```

```
            ;2420  .PAGE
            ;2421  .TOC "   SELECT LOWER CONTROLLER"
            ;2422 ;*****************************************************************
            ;2423 ;++
            ;2424 ;
            ;2425 ; Functional Description:
            ;2426 ; ----------------------
            ;2427 ;
            ;2428 ; This subroutine can be called to select the lower
            ;2429 ; Controller on a MS780-E/H.
            ;2430 ; If the Lower controller is misconfigured then a
            ;2431 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2432 ;
            ;2433 ; Calling Constraint: =00
            ;2434 ; ------------------
            ;2435 ;
            ;2436 ;
            ;2437 ; Inputs:
            ;2438 ; ------
            ;2439 ;
            ;2440 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2441 ;
            ;2442 ; Outputs:
            ;2443 ; -------
            ;2444 ;
            ;2445 ; RC[0D] will have the address of CNFG Register C
            ;2446 ;        ( 2000x008 )
            ;2447 ;
            ;2448 ; Side Effects:
            ;2449 ; ------------
            ;2450 ;
            ;2451 ; Contents of the following registers will lost:
            ;2452 ;
            ;2453 ; D
            ;2454 ;
            ;2455 ; The Lower controller on the MS780-E/H will be configured
            ;2456 ; when the subroutine is exitted with a RETURN2
            ;2457 ;--
            ;2458 ;*****************************************************************
            ;2459 =00
            ;2460 SELECT.LOWER:
            ;2461  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1020, 0818,0039,1980,0800,0000,119E  ;2462   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1021, 0000,003E,0180,0800,0000,0001  ;2463  RETURN1    ; Lower Controller Misconfigured
U 1022, 0810,0038,0180,0970,0000,1023  ;2464  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;2465  RC[0D]_D+K[.8],    ; Set the address of CNFG Reg D
U 1023, 0019,0016,0180,09E8,0000,0002  ;2466   RETURN2   ; Let caller know that the controller
            ;2467       ; ...was configured properly
```

```
;2468  .PAGE
;2469  .TOC "   SELECT UPPER CONTROLLER"
;2470 ;*************************************************************
;2471 ;++
;2472 ;
;2473 ; Functional Description:
;2474 ; ----------------------
;2475 ;
;2476 ; This subroutine can be called to select the upper
;2477 ; Controller on a MS780-E/H.
;2478 ; If the Upper controller is misconfigured then a
;2479 ; RETURN1 will be made.  Otherwise a RETURN2
;2480 ;
;2481 ; Calling Constraint: =00
;2482 ; ------------------
;2483 ;
;2484 ;
;2485 ; Inputs:
;2486 ; ------
;2487 ;
;2488 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2489 ;
;2490 ; Outputs:
;2491 ; -------
;2492 ;
;2493 ; RC[0D] will have the address of CNFG Register D
;2494 ;        ( 2000x010 )
;2495 ;
;2496 ; Side Effects:
;2497 ; ------------
;2498 ;
;2499 ; Contents of the following registers will lost:
;2500 ;
;2501 ; D
;2502 ;
;2503 ; The Upper controller on the MS780-E/H will be configured
;2504 ; when the subroutine is exitted with a RETURN2
;2505 ;--
;2506 ;*************************************************************
;2507 =00
;2508 SELECT.UPPER:
;2509  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,0980,0800,0000,119E  ;2510   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2511  RETURN1    ; Upper Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2512  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2513  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 102B, 0019,0016,8580,09E8,0000,0002  ;2514   RETURN2   ; Let caller know that the controller
;2515       ; ...was configured properly
```

```
            ;2516  .PAGE
            ;2517  .TOC "   Set Diagnostic Mode Routine"
            ;2518 ;++
            ;2519 ; FUNCTIONAL DESCRIPTION:
            ;2520 ;
            ;2521 ; This routine is used to set the specified diagnostic mode
            ;2522 ; in Register B. Other bits in the register remain unchanged.
            ;2523 ;
            ;2524 ; CALLING CONSTRAINT:
            ;2525 ;
            ;2526 ; =0
            ;2527 ;
            ;2528 ; INPUTS:
            ;2529 ;
            ;2530 ; d - Contains the mode to set in bits<1:0>
            ;2531 ; rc[0e] - Contains base address of controller
            ;2532 ;
            ;2533 ; SIDE EFFECTS:
            ;2534 ;
            ;2535 ; d,va,sc are destroyed
            ;2536 ;--
            ;2537 SET_DIAG_MODE:
U 11AF, 0010,0038,D9F8,0970,0284,71B0  ;2538  va_rc[0e],sc_k[.9],q_0 ; Set address of Register B
U 11B0, 0D00,003C,0180,0803,0000,11B1  ;2539  va_va+4,d_dal.sc ; Shift specified mode into position
U 11B1, 0000,003C,01E0,0800,0000,11B2  ;2540  q_d   ; Save the diagnostic mode bits
U 11B2, 0000,003C,0180,C800,0000,11B3  ;2541  d[long]_cache.p  ; Read the contents of the CNFG register B
U 11B3, 081D,0030,0180,0800,0000,11B4  ;2542  d_d.or.q  ; Set the diagnostic mode bits
U 11B4, 0000,003E,0180,A800,0000,0001  ;2543  cache.p_d[long],return1 ; Set the specified mode and return
            ;2544
```

```
            ;2545  .PAGE
            ;2546  .TOC "   SBI Unjam Routine"
            ;2547 ;++
            ;2548 ; FUNCTIONAL DESCRIPTION:
            ;2549 ;
            ;2550 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2551 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2552 ; and finally HOLD for the last 16 cycles.
            ;2553 ;
            ;2554 ; CALLING CONSTRAINT:
            ;2555 ;
            ;2556 ; =0
            ;2557 ;
            ;2558 ; SIDE EFFECTS:
            ;2559 ;
            ;2560 ; D Reg destroyed
            ;2561 ;--
            ;2562   ;
            ;2563   ; assert hold for 16 cycles
            ;2564   ;
            ;2565 SBI.UNJAM:
            ;2566  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 11B5, 0818,0038,6580,8000,0010,1072  ;2567      clk.ubcc  ; set micro cc's for next cycle
            ;2568 =0
            ;2569 SBI.UNJAM.1:
            ;2570  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2571       clk.ubcc,z?, ; ...
U 1072, 0819,0100,0580,8000,0010,1072  ;2572      j/sbi.unjam.1 ; test for completion
            ;2573    ;
            ;2574    ; assert hold and unjam for 16 cycles
            ;2575    ;
            ;2576  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 1073, 0818,0038,6580,8800,0010,1074  ;2577      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2578 =0
            ;2579 SBI.UNJAM.2:
            ;2580  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2581       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1074, 0819,0100,0580,8800,0010,1074  ;2582      z?,j/sbi.unjam.2 ; ...
            ;2583    ;
            ;2584    ; assert hold for 16 cycles
            ;2585    ;
            ;2586  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1075, 0818,0038,6580,8000,0010,1076  ;2587      clk.ubcc  ; and set cc's for next cycle
            ;2588 =0
            ;2589 SBI.UNJAM.3:
            ;2590  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2591       clk.ubcc,z?, ; ...
U 1076, 0819,0100,0580,8000,0010,1076  ;2592      j/sbi.unjam.3 ; ...
U 1077, 0000,003E,0180,0800,0000,0001  ;2593  return1   ; exit
```

```
            ;2594  .PAGE
            ;2595  .TOC "   RESET SUBTEST NUMBER"
            ;2596 ;++
            ;2597 ; FUNCTIONAL DESCRIPTION:
            ;2598 ;
            ;2599 ; Since some  of the tests will have to be restarted when two
            ;2600 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2601 ; the subtest counter to zero ( only for thoes tests that have
            ;2602 ; more than one subtest). This subroutine may also be necessary
            ;2603 ; when a test is being loop on.
            ;2604 ;
            ;2605 ; CALLING CONSTRAINT:
            ;2606 ;
            ;2607 ; =0
            ;2608 ; SIDE EFFECTS:
            ;2609 ;
            ;2610 ; D is destroyed
            ;2611 ;
            ;2612 ;--
            ;2613 RESET.SUBTST:
            ;2614  RC[0F]_0,    ; Reset subtest number
U 11B6, 0003,003E,0180,09F8,0000,0001  ;2615   RETURN1    ; ...and return
```

```
            ;2616  .PAGE
            ;2617  .TOC "   Initialize the SBI"
            ;2618 ;++
            ;2619 ; FUNCTIONAL DESCRIPTION:
            ;2620 ;
            ;2621 ; This routine performs the following functions:
            ;2622 ;
            ;2623 ;  Executes an SBI Unjam
            ;2624 ;  Clears the SBI Fault Latch
            ;2625 ;  Clears the SBI Error Register
            ;2626 ;
            ;2627 ; CALLING CONSTRAINT:
            ;2628 ;
            ;2629 ; =0
            ;2630 ;
            ;2631 ; SIDE EFFECTS:
            ;2632 ;
            ;2633 ; D & Q Register destroyed
            ;2634 ;--
            ;2635 =0
            ;2636 SBI.INIT:
U 1078, 0000,003D,0180,0800,0000,11B5  ;2637  call[sbi.unjam]  ; Unjam the SBI
U 1079, 0818,0038,C180,0800,0000,11B7  ;2638  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11B7, 0801,0028,6580,3C00,0000,11B8  ;2639  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11B8, 0F00,003C,6D80,3C00,0000,11B9  ;2640  id[fault]_d,d_0
U 11B9, 0000,003C,6D80,3C00,0000,11BA  ;2641  id[fault]_d
U 11BA, 0000,003E,6580,3C00,0000,0001  ;2642  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
                ;2643  .PAGE
                ;2644  .TOC "   Disable Cache Routine"
                ;2645 ;++
                ;2646 ; FUNCTIONAL DESCRIPTION:
                ;2647 ;
                ;2648 ; This routine disables cache hits by setting bits <16:15>
                ;2649 ; in the maintenance register.
                ;2650 ;
                ;2651 ; CALLING SEQUENCE:
                ;2652 ;
                ;2653 ; =0
                ;2654 ;
                ;2655 ; SIDE EFFECTS:
                ;2656 ;
                ;2657 ; D & Q Registers destroyed
                ;2658 ;--
                ;2659 DISABLE.CACHE:
U 11BB, 0818.0038.4580.0800.0000.11BC  ;2660  d_k[.8000]  ; ... and seed constant
U 11BC, 0500.003C.0180.0800.0000.11BD  ;2661  d_d.left  ; Generate final constant
U 11BD, 0819.0030.4580.0800.0000.11BE  ;2662  d_d.or.k[.8000]  ; ... = 00018000
U 11BE, 0000.003E.7580.3C00.0000.0001  ;2663  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2664  .PAGE
            ;2665  .TOC "   Enable Cache"
            ;2666 ;++
            ;2667 ; FUNCTIONAL DESCRIPTION:
            ;2668 ;
            ;2669 ; This routine enables cache group 0 by clearing the force
            ;2670 ; miss bit in the maintenance register.
            ;2671 ;
            ;2672 ; CALLING CONSTRAINT:
            ;2673 ;
            ;2674 ; =0
            ;2675 ;
            ;2676 ; SIDE EFFECTS:
            ;2677 ;
            ;2678 ; D, Q AND SC Registers destroyed
            ;2679 ;--
            ;2680 ENABLE.CACHE:
U 11BF, 0000.003C.75F0.2C00.0000.11C0  ;2681  q_id[maint]  ; Get current maintenance register
U 11C0, 0000.003C.6580.0800.0084.71C1  ;2682  sc_k[.10]  ; Setup to clear bit 16
U 11C1, 0801.2034.0180.0800.0000.11C2  ;2683  d_q.and.mask  ; Clear bit 16
U 11C2, 0000.003E.7580.3C00.0000.0001  ;2684  id[maint]_d,return1 ; Rewrite register and return
            ;2685
```

```
            ;2686  .PAGE
            ;2687  .TOC "   CHECK UTRAP"
            ;2688 ;++
            ;2689 ;FUNCTIONAL DESCRIPTION:
            ;2690 ;
            ;2691 ; This subroutine is used to check wether a Utrap occured.
            ;2692 ; It takes the contents of the Save Utrap flags register
            ;2693 ; R[0E], and compares them to the contrnts of the Utrap
            ;2694 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2695 ; according to the results of the test  (i.e., contents of
            ;2696 ; these registers are equal or not).
            ;2697 ; CALLING CONSTRAINT:
            ;2698 ;
            ;2699 ; =00
            ;2700 ;
            ;2701 ; INPUTS:
            ;2702 ;
            ;2703 ; R[0E]- Saved Utrap Mask
            ;2704 ; R[0F]- Expected Utrap Mask
            ;2705 ;
            ;2706 ;--
            ;2707 CHECK_UTRAP:
U 11C3. 0800.003C.0180.0A70.0000.11C4  ;2708  D_R[0E]   ; Reg D is loaded with the trap mask
U 11C4. 0001.003C.0180.09C0.0000.11C5  ;2709  RC[08]_D  ; Save expected Utrap flag for error logout
U 11C5. 0800.003C.0180.0A78.0000.11C6  ;2710  D_R[0F]   ; Get recieved Utrap flag to D reg
U 11C6. 0001.003C.0180.09B8.0000.11C7  ;2711  RC[07]_D  ; Save in RC[07]
            ;2712  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 11C7. 000D.0020.0180.0A70.0010.11C8  ;2713        CLK.UBCC
U 11C8. 0003.013C.0180.0AF8.0000.107A  ;2714  Z?,R[0F]_0  ; Branch if no traps
U 107A. 0000.003E.0180.0800.0000.0002  ;2715 =0 RETURN2   ; Utrap occured
U 107B. 0000.003E.0180.0800.0000.0001  ;2716  RETURN1   ; No Utrap  return
            ;2717 ;
```

```
            ;2718  .PAGE
            ;2719  .TOC "   Set Trap Mask"
            ;2720 ;++
            ;2721 ; FUNCTIONAL DESCRIPTION:
            ;2722 ;
            ;2723 ; This routine is used to set a bit in the Micro Trap Mask
            ;2724 ; R[0F].
            ;2725 ;
            ;2726 ; CALLING CONSTRAINT:
            ;2727 ;
            ;2728 ; =0
            ;2729 ;
            ;2730 ; INPUTS:
            ;2731 ;
            ;2732 ; SC - Contains bit number to set.
            ;2733 ;
            ;2734 ; OUTPUTS:
            ;2735 ;
            ;2736 ; rc[0E] - Contains the current trap mask
            ;2737 ;
            ;2738 ; SIDE EFFECTS:
            ;2739 ;
            ;2740 ; d regsiter is destroyed
            ;2741 ;--
            ;2742 SET_TRAP_MASK:
U 11C9, 0800,003C,0180,0A78,0000,11CA  ;2743  d_r[0f]
U 11CA, 0801,001C,0180,0AF8,0000,11CB  ;2744  r[0f]_d.ornot.mask,d_alu ; Set mask
U 11CB, 0001,003E,0180,0AF0,0000,0001  ;2745  r[0E]_d,return1    ; Save mask and return
```

```
                ;2746  .PAGE
                ;2747  .TOC "   Find MS780-E Memory"
                ;2748 ;**
                ;2749 ; FUNCTIONAL DESCRIPTION:
                ;2750 ;
                ;2751 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
                ;2752 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
                ;2753 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
                ;2754 ;     3D will hold the I/O base address of the MS780-E subsystems found
                ;2755 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
                ;2756 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
                ;2757 ;     first MS780-E found will have  its  starting  address  set  to  0
                ;2758 ;     (zero).
                ;2759 ;     All the other MS780-E/H's found will have their starting  address
                ;2760 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
                ;2761 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
                ;2762 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
                ;2763 ;     current MS780-E unit under test.
                ;2764 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
                ;2765 ;     set to maximum.
                ;2766 ;
                ;2767 ; CALLING CONSTRAINT:
                ;2768 ;
                ;2769 ; =00
                ;2770 ;
                ;2771 ; OUTPUTS:
                ;2772 ;
                ;2773 ; rc[0e] - Contains address of Configuration Register
                ;2774 ; r[0]  - Contains TR level
                ;2775 ;
                ;2776 ; SIDE EFFECTS:
                ;2777 ;
                ;2778 ; D register is destroyed.
                ;2779 ;--
                ;2780 =0
                ;2781 FIND.MS780E.AND.UBA:
U 107C. 0000.003D.0180.0800.0000.1078  ;2782  call[sbi.init]  ; Make sure SBI is enabled
U 107D. 0003.003C.0180.0988.0000.11CC  ;2783  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11CC. 0818.0038.1980.0800.0000.11CD  ;2784  d_k[zero]  ; Clear MS780-E/H counter
U 11CD. 0000.003C.F980.3C00.0000.11CE  ;2785  id[3e]_d  ; ...
U 11CE. 0000.003C.9180.3C00.0000.11CF  ;2786  ID[24]_D  ; Clear
U 11CF. 0018.0038.0580.0A80.0000.11D0  ;2787  r[0]_k[.1]  ; Init TR counter
U 11D0. 0F03.003C.0180.09F0.0000.107E  ;2788  d_0,rc[0E]_0  ; Clear "Save" location for
                ;2789      ; ...CNFG A Reg
                ;2790 =0
                ;2791 FIND.MEM.LOOP:
U 107E. 0800.003D.0180.0A00.0000.120C  ;2792  d_r[0],call[generate.io]; Generate address of TR level
U 107F. 0001.003C.0180.09F0.0200.1080  ;2793  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1080. 0000.003D.8980.0800.0084.71C9  ;2794 =0 set.trap.mask[.d] ; Enable timeout micro traps
                ;2795  d[long]_cache.p, ; Read the specified tr level
U 1081. 0000.003C.0180.C970.0000.11D1  ;2796   ic_rc[0e]  ; ...
U 11D1. 0000.003C.0180.0A78.0010.11D2  ;2797  alu_r[0f],clk.ubcc ; Check for no response
U 11D2. 0001.013C.0180.0880.0082.1082  ;2798  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 1082. 0000.003C.0180.0800.0000.11D3  ;2799 =0 j/check.adpt.code ; Check for a memory
U 1083. 0000.003C.0180.0800.0000.1204  ;2800  j/find.mem.lp1  ; Try next tr
```

```
            ;2801 CHECK.ADPT.CODE:
U 11D3, 0000,003C,1D80,0800,1404,71D4   ;2802  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11D4, 0000,163C,0180,0800,0000,1018   ;2803  state7-4?  ; Branch on the adapter type
U 1018, 0000,003C,8980,0800,0084,71D5   ;2804 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1019, 0000,003C,8980,0800,0084,71D6   ;2805  j/ms780.c,sc_k[.d] ; MS780-C
U 101A, 0000,003C,0180,0800,0000,11DF   ;2806  j/CHECK.FOR.UBA  ; Could be a UBA
U 101B, 0000,003C,0180,0800,0000,1204   ;2807  j/find.mem.lp1  ; Not a memory
U 101C, 0000,003C,6580,0800,0084,71DB   ;2808  j/ma780.max,sc_k[.10] ; MA780
U 101D, 0000,003C,0180,0800,0000,1204   ;2809  j/find.mem.lp1  ; Not a memory
U 101E, 0010,0038,0180,09F0,0000,11F1   ;2810  rc[0e]_lc,j/found.ms780e ; MS780-E
U 101F, 0010,0038,0180,09F0,0000,11F1   ;2811  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2812 ;
            ;2813 ; Found an MS780-A or -C. Set the starting address
            ;2814 ; to maximum.
            ;2815 ;
U 11D5, 0818,0038,5D80,0800,0000,11D7   ;2816 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11D6, 0818,0038,0580,0800,0000,11D7   ;2817 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2818 MS780.COMMON:
U 11D7, 0819,0030,7180,0800,0000,11D8   ;2819  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11D8, 0000,003C,01F8,0803,0080,D1D9   ;2820  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11D9, 0D00,003C,0180,0800,0000,11DA   ;2821  d_dal.sc  ; Put data in bits<29:14>
            ;2822  cache.p_d[long], ; Set the starting address to maximum
U 11DA, 0000,003C,0180,A800,0000,1204   ;2823       j/find.mem.lp1 ; and continue TR scan
            ;2824 ;
            ;2825 ; Found an MA780. Set the starting address to maximum
            ;2826 ;
            ;2827 MA780.MAX:
U 11DB, 0818,0038,39F8,0803,0000,11DC   ;2828  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11DC, 0D00,003C,0180,0803,0000,11DD   ;2829  d_dal.sc,va_va+4 ; Put in proper position
U 11DD, 0000,003C,0180,0803,0000,11DE   ;2830  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2831  cache.p_d[long], ; Set starting address to maximum
U 11DE, 0000,003C,0180,A800,0000,1204   ;2832       j/find.mem.lp1
            ;2833 ;
            ;2834 ; Found a UBA or MBA. Check which it is
            ;2835 ;
            ;2836 ;
            ;2837 CHECK.FOR.UBA:
U 11DF, 0000,003C,91F0,2C00,0000,11E0   ;2838  Q_ID[24]  ; Get Fond UBA Flag
U 11E0, 0001,203C,0180,0800,0010,11E1   ;2839  ALU_Q,CLK.UBCC  ; Is it 0?
U 11E1, 0000,013C,0180,0800,0000,1084   ;2840  Z?
U 1084, 0000,003C,0180,0800,0000,1204   ;2841 =0 J/FIND.MEM.LP1  ; Already found one UBA
U 1085, 0818,0038,1D80,0800,0000,11E2   ;2842  D_SC   ; Get adapter code
U 11E2, 0000,193C,0180,0800,0000,1027   ;2843  D3?   ; Is bit 3 set?
U 1027, 0000,003C,0180,0800,0000,1204   ;2844 =0111 J/FIND.MEM.LP1  ; Not a UBA. Continue search.
U 102F, 0810,0038,0180,0970,0000,11E3   ;2845  D_RC[0E]  ; Get I/O Base Address of the UBA
U 11E3, 0000,003C,9180,3C00,0000,11E4   ;2846  ID[24]_D  ; Save it in ID Reg 24 and at the same
            ;2847      ; ...time set FOUND UBA FLAG
U 11E4, 0819,0030,31E0,0800,0000,11E5   ;2848  D_D.OR.K[.40],Q_D ; Generate Address of DPR1
U 11E5, 0819,0030,1180,0800,0000,11E6   ;2849  D_D.OR.K[.4]  ; ...D = 2000x044
U 11E6, 0000,003C,A180,3C00,0000,11E7   ;2850  ID[28]_D  ; Save in ID Reg 28
U 11E7, 08B8,0038,4180,0800,0000,11E8   ;2851  D_K[.80].LEFT3  ; Generate now address of MR0
U 11E8, 0500,003C,0180,0800,0000,11E9   ;2852  D_D.LEFT  ; ...D = 800
U 11E9, 081D,0030,0180,0800,0000,11EA   ;2853  D_D.OR.Q  ; ...D = 2000x800
U 11EA, 0000,003C,9580,3C00,0000,11EB   ;2854  ID[25]_D  ; Save in ID Reg 25
U 11EB, 0818,0038,1D80,0800,0000,11EC   ;2855  D_SC   ; Get adapter code
```

```
U 11EC, 0000,003C,6580,0800,0084,71ED  ;2856  SC_K[.10]  ; Prepare to shift adapter code to generate
        ;2857     ; ...UBA address space (based on UBA #)
U 11ED, 0000,003C,0980,0800,0084,91EE  ;2858  SC_SC+K[.2]  ; ...SC = x12
        ;2859  D_D.AND.K[.3],  ; Mask UBA number (bits <1:0>)
U 11EE, 0819,0034,0DF8,0800,0000,11EF  ;2860   Q_0  ; Clear the Q Reg
U 11EF, 0D00,003C,0180,0800,0000,1086  ;2861  D_DAL.SC  ; Shift
        ;2862 =0 D_K[.80],  ; Generate x20100000 in the D reg
U 1086, 0818,0039,4180,0800,0000,120C  ;2863   CALL[GENERATE.IO]
U 1087, 081D,0030,0180,0800,0000,11F0  ;2864  D_D.OR.Q  ; D = x201n0000 (where n = 0, 4, 8 or C)
        ;2865  ID[26]_D,  ; Save in ID Reg 26
U 11F0, 0000,003C,9980,3C00,0000,1204  ;2866   J/FIND.MEM.LP1 ; Continue searching for MS780 E's
        ;2867
        ;2868 ;
        ;2869 ; Found an MS780E
        ;2870 ;
        ;2871 FOUND.MS780E:
U 11F1, 0000,003C,F9F0,2C00,0000,11F2  ;2872  q_id[3e]  ; Set up to see how many MS780-E's
        ;2873  alu_q.xor.k[.3], ; Are there Maximum units?
U 11F2, 0019,2020,0D80,0800,0010,11F3  ;2874   clk.ubcc ; Check condition codes
U 11F3, 0000,013C,0180,0800,0000,1088  ;2875  z?   ; Are we at maximum units for system
U 1088, 0000,003C,0180,0800,0000,11F4  ;2876 =0 J/ms780e.tst  ; No go find how many units ther are
U 1089, 0818,0038,C180,0800,0000,108A  ;2877  d_k[.ffff]  ; Yes set address to maximum
U 108A, 0000,003D,0180,0800,0000,1210  ;2878 =0 call[set.start.addr] ; .....
U 108B, 0000,003C,0180,0800,0000,1204  ;2879  j/find.mem.lp1  ; Go check to see if we are done
        ;2880 ;
        ;2881 MS780E.TST:
        ;2882  alu_rc[01],  ; Check "Found MS780E Flag"
U 11F4, 0010,0038,0180,0908,0010,11F5  ;2883   clk.ubcc ; Check condition codes
U 11F5, 0810,0138,0180,0970,0000,108C  ;2884  z?,d_rc[0e]  ; Is this unit the first MS780E ?
        ;2885 =0 j/not.first.ms780e, ; No
U 108C, 0818,0038,C180,0800,0000,1090  ;2886   d_k[.ffff] ; Set starting address
U 108D, 0001,003C,0180,0988,0000,11F6  ;2887  rc[01]_d  ; Yes put base address in rc[01]
U 11F6, 0818,0038,7980,0800,0000,11F7  ;2888  d_k[.30]  ; Set up SC to point to first unit
U 11F7, 0019,0014,F580,0800,0082,11F8  ;2889  alu_d+k[.a],sc_alu ; ...
U 11F8, 0810,0038,0180,0908,0000,11F9  ;2890  d_rc[01]  ; Put base address of unit in D
U 11F9, 0000,003C,0180,3400,0000,11FA  ;2891  id(sc)_d  ; Put base address in ID pointed to by SC
U 11FA, 0F00,003C,0180,0800,0000,108E  ;2892  d_0   ; Prepare to set starting address to Zero
U 108E, 0000,003D,0180,0800,0000,1210  ;2893 =0 call[set.start.addr] ; Go do it
        ;2894  j/find.mem.lp1,  ; Check to see if we are done
U 108F, 0003,003C,0180,09E8,0000,1204  ;2895   rc[0d]_0 ; ....
        ;2896 =0
        ;2897 NOT.FIRST.MS780E:
U 1090, 0000,003D,0180,0800,0000,1210  ;2898  call[set.start.addr] ; Go set starting address to maximum
U 1091, 0000,003C,F9F0,2C00,0000,11FB  ;2899  q_id[3e]  ; Get the number of MS780-Es found
U 11FB, 0819,2014,0580,0800,0000,11FC  ;2900  d_q+k[.1]  ; Increment this counter
U 11FC, 0000,003C,F980,3C00,0000,11FD  ;2901  id[3e]_d  ; Save the counter
U 11FD, 0018,0038,11C0,0800,0000,11FE  ;2902  q_k[.4]   ; Prepare to form pointer to the ID registers
        ;2903     ; ...were the I/O base addresses will be stored
U 11FE, 081D,2000,7980,0800,0000,11FF  ;2904  d_q-d,k[.30]  ; ... adjust pointer
U 11FF, 0819,0014,7980,0800,0000,1200  ;2905  d_d+k[.30]  ; Point the SC to the ID register
U 1200, 0000,003C,F580,0800,0000,1201  ;2906  k[.a]   ; ...to receive I/O base address
U 1201, 0019,0014,F580,0800,0082,1202  ;2907  alu_d+k[.a],sc_alu ; ...
U 1202, 0810,0038,0180,0970,0000,1203  ;2908  d_rc[0e]  ; Get base address to d
U 1203, 0000,003C,0180,3400,0000,1204  ;2909  id(sc)_d  ; Move base address to ID pointed to by SC
        ;2910
```

```
            ;2911 ;
            ;2912 ; Try next tr
            ;2913 ;
            ;2914 FIND.MEM.LP1:
U 1204, 0000.003C.0180.0880.0000.1205  ;2915  LA_RA[0]  ; Get TR level
U 1205, 0818.0014.0580.0A80.0000.1206  ;2916  r[0]_la+k[.1],d_alu ; Increment TR level
            ;2917  alu_d.and.k[.f],
U 1206, 0019.0034.6180.0800.0010.1207  ;2918       clk.ubcc  ; Check if all done
U 1207, 0000.013C.0180.0800.0000.1092  ;2919  z?    ; Branch if yes
U 1092, 0000.003C.0180.0800.0000.107E  ;2920 =0 j/find.mem.loop  ; Try next TR
U 1093, 0810.0038.0180.0908.0010.1208  ;2921  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1208, 0000.013C.0180.0800.0000.1094  ;2922  z?    ; Check condition codes
            ;2923 =0 J/UBA.FOUND,   ; Yes MS780E was found
U 1094, 0001.003C.0180.09F0.0000.1209  ;2924   rc[0e]_d
U 1095, 0000.003E.0180.0800.0000.0001  ;2925  return1    ; No MS780E found
            ;2926 UBA.FOUND:
U 1209, 0000.003C.91F0.2C00.0000.120A  ;2927  Q_ID[24]  ; Get UBA FOUND FLAG
U 120A, 0001.203C.0180.0800.0010.120B  ;2928  ALU_Q,CLK.UBCC  ; Did we find a UBA?
U 120B, 0000.013C.0180.0800.0000.1096  ;2929  Z?    ; ...
U 1096, 0000.003E.0180.0800.0000.0002  ;2930 =0 RETURN2   ; OK. At this point we found a MS780-E/H and
            ;2931      ; ...a UBA
U 1097, 0000.003E.0180.0800.0000.0001  ;2932  RETURN1    ; NO UBA FOUND.
            ;2933
```

```
            ;2934  .PAGE
            ;2935  .TOC "   Generate IO Address"
            ;2936 ;++
            ;2937 ; FUNCTIONAL DESCRIPTION:
            ;2938 ;
            ;2939 ; This routine is used to generate an SBI Configuration Register
            ;2940 ; address from a TR level.
            ;2941 ;
            ;2942 ; CALLING CONSTRAINT:
            ;2943 ;
            ;2944 ; =0
            ;2945 ;
            ;2946 ; INPUTS:
            ;2947 ;
            ;2948 ; D - Contains TR level
            ;2949 ;
            ;2950 ; OUTPUTS:
            ;2951 ;
            ;2952 ; D - Contains SBI IO address
            ;2953 ;--
            ;2954 GENERATE.IO:
U 120C, 0000,003C,89F8,0800,0084,720D  ;2955  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 120D, 0D00,003C,8D80,0800,0084,720E  ;2956  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 120E, 0000,003C,0980,0800,0084,B20F  ;2957  sc_sc-k[.2]  ; insert bit 29
U 120F, 0801,001E,0180,0800,0000,0001  ;2958  d_d.ornot.mask,return1 ; and return
```

```
            ;2959  .PAGE
            ;2960  .TOC "   Set Starting Address"
            ;2961 ;**
            ;2962 ; FUNCTIONAL DESCRIPTION:
            ;2963 ;
            ;2964 ; This routine is used to set the starting address of the
            ;2965 ; memory to the specified value.
            ;2966 ;
            ;2967 ; CALLING CONSTRAINT:
            ;2968 ;
            ;2969 ; =0
            ;2970 ;
            ;2971 ; INPUTS:
            ;2972 ;
            ;2973 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2974 ;     (B Register Image divided by 2**16)
            ;2975 ; rc[0e] - Contains Address of Register A
            ;2976 ;
            ;2977 ; OUTPUTS:
            ;2978 ;
            ;2979 ; d - Contains aligned starting address (B Register Image)
            ;2980 ;
            ;2981 ; SIDE EFFECTS:
            ;2982 ;
            ;2983 ; d,q,va, and sc are destroyed
            ;2984 ;--
            ;2985 SET.START.ADDR:
U 1210, 0010,0038,6580,0970,0284,7211  ;2986  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1211, 0000,003C,01F8,0803,0000,1212  ;2987  va_va+4,q_0  ; Set address to Register B
            ;2988  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1212, 0D00,003C,0980,0800,0084,B213  ;2989      sc_sc-k[.2]  ; Setup to insert bit 14
U 1213, 0801,001C,0180,0800,0000,1214  ;2990  d_d.ornot.mask  ; Insert Write Enable bit
U 1214, 0000,003E,0180,A800,0000,0001  ;2991  cache.p_d[long],return1 ; Set the starting address and return
            ;2992
```

```
            ;2993  .PAGE
            ;2994  .TOC "   ENABLE NEXT MS780-E"
            ;2995 ;++
            ;2996 ; FUNCTIONAL DESCRIPTION:
            ;2997 ;
            ;2998 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2999 ;    They will be tested  separately, according to the TR level at which
            ;3000 ;    they are located,  starting with the one  at the lowest level first.
            ;3001 ;          When  a test has finished  with the first unit, it will have to call
            ;3002 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;3003 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;3004 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;3005 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;3006 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;3007 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;3008 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;3009 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;3010 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;3011 ;
            ;3012 ; CALLING CONSTRAINT:
            ;3013 ;
            ;3014 ;  =00
            ;3015 ;
            ;3016 ;--
            ;3017 ENABLE.NEXT.MS780E:
U 1215, 0000,003C,F9F0,2C00,0000,1216  ;3018  Q_ID[3E]  ; Get total number of MS780E to Q
            ;3019  ALU_Q,   ; Check to see if it is zero
U 1216, 0001,203C,0180,0800,0010,1217  ;3020      CLK.UBCC  ; Check Condition Codes
U 1217, 0000,013C,0180,0800,0000,1098  ;3021  Z?   ; ...for zero
U 1098, 0000,003C,0180,0800,0000,1218  ;3022 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1099, 0000,003E,0180,0800,0000,0002  ;3023  RETURN2   ; Zero No more units to test
            ;3024 ENABLE.NEXT:
U 1218, 0818,0038,C180,0800,0000,109A  ;3025  D_K[.FFFF]  ; Load D with Maximum Starting address
U 109A, 0000,003D,0180,0800,0000,1210  ;3026 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 109B, 0818,0038,7980,0800,0000,1219  ;3027  D_K[.30]  ; Prepare to point to the ID register holding
            ;3028      ; ...the I/O Base address of the next MS780-E
U 1219, 0819,0014,1180,0800,0000,121A  ;3029  D_D+K[.4]  ; ...
U 121A, 0000,003C,F580,0800,0000,121B  ;3030  K[.A]   ; ...
U 121B, 0819,0014,F580,0800,0000,121C  ;3031  D_D+K[.A]  ; ...
U 121C, 0000,003C,F9F0,2C00,0000,121D  ;3032  Q_ID[3E]  ; Get MS780-E Counter
            ;3033  D_D-Q,   ; D now holds the pointer to the ID register
U 121D, 081D,0000,0180,0800,0082,121E  ;3034   SC_ALU  ; ...
U 121E, 0000,003C,01F0,2400,0000,121F  ;3035  Q_ID(SC)  ; Q gets adrress of next MS780E
U 121F, 0001,203C,0180,09F0,0000,1220  ;3036  RC[0E]_Q  ; Save it also in RC[0E]
U 1220, 0F03,003C,0180,09E8,0000,109C  ;3037  RC[0D]_0,D_0  ; Set starting address to zero
U 109C, 0000,003D,0180,0800,0000,1210  ;3038 =0 CALL[SET.START.ADDR] ; ....
U 109D, 0F00,003C,F9F0,2C00,0000,1221  ;3039  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1221, 081D,2008,0180,0800,0000,1222  ;3040  D_Q-D-1   ; Subtract 1 from total
U 1222, 0000,003C,F980,3C00,0000,109E  ;3041  ID[3E]_D  ; Adjust the pointer
U 109E, 0000,003D,0180,0800,0000,11B6  ;3042 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 109F, 0000,003E,0180,0800,0000,0001  ;3043  RETURN1   ; Tell caller another unit was found
            ;3044
```

```
            ;3045  .PAGE
            ;3046  .TOC "   Set ECC Bits"
            ;3047 ;++
            ;3048 ; FUNCTIONAL DESCRIPTION:
            ;3049 ;
            ;3050 ; This routine is used to set the specified ECC bits
            ;3051 ; in Register B. Other bits in the register remain unchanged.
            ;3052 ;
            ;3053 ; CALLING CONSTRAINT:
            ;3054 ;
            ;3055 ; =0
            ;3056 ;
            ;3057 ; INPUTS:
            ;3058 ;
            ;3059 ; d - Contains the bits to set in <6:0>
            ;3060 ; rc[0] - Contains base address of controller
            ;3061 ;
            ;3062 ; SIDE EFFECTS:
            ;3063 ;
            ;3064 ; d,va,sc are destroyed
            ;3065 ;--
            ;3066 SET_ECC:
U 1223, 0010,0038,0180,0970,0200,1224  ;3067  va_rc[0E]  ; Set address of Register B
U 1224, 0000,003C,0180,0803,0000,1225  ;3068  va_va+4    ; ...in VA
U 1225, 0000,003C,01E0,C800,0000,1226  ;3069  q_d,d[long]_cache.p ; Read current contents of register
U 1226, 0819,0024,5980,0800,0000,1227  ;3070  d_d.andnot.k[.7f] ; Clear current ECC bits
U 1227, 081D,0030,0180,0800,0000,1228  ;3071  d_d.or.q  ; Insert new ecc bits
U 1228, 0000,003E,0180,A800,0000,0001  ;3072  cache.p_d[long],return1 ; Set the specified bits and return
            ;3073
            ;3074
```

```
;3075  .PAGE
;3076  .TOC "     MS780-E/H CNFG REG CLEANUP"
;3077 ;*************************************************************
;3078 ;++
;3079 ;
;3080 ; Functional Description:
;3081 ; ----------------------
;3082 ;
;3083 ;       This subroutine is used to clear all error conditions
;3084 ; in the MS780-E/H Configuration/Status/Error Registers.
;3085 ; Bits beeing cleared are:
;3086 ;
;3087 ; CNFG Regsiter B <11:00>
;3088 ;
;3089 ; CNFG Register C and D <31:28> and <10:06>
;3090 ;   ---
;3091 ;
;3092 ;
;3093 ; Calling Constraint: =0
;3094 ; ------------------
;3095 ;
;3096 ;
;3097 ; Inputs:
;3098 ; ------
;3099 ;
;3100 ; RC[0E] MUST contain the I/O base address of
;3101 ; the MS780-E/H currently under test
;3102 ;
;3103 ; Outputs:
;3104 ; -------
;3105 ;
;3106 ; NO specific outputs.
;3107 ; Above mentioned bits will be cleared.
;3108 ;
;3109 ; Side Effects:
;3110 ; ------------
;3111 ;
;3112 ; The contents of the following registers will be lost
;3113 ; Q, D, SC, VA
;3114 ;
;3115 ;
;3116 ;--
;3117 ;*************************************************************
;3118 MS780E.CLEANUP:
U 1229, 0010,0038,0180,0970,0200,122A  ;3119  VA_RC[0E]    ; Point to CNFG Reg A of MS780-E/H
        ;3120  D_0,      ; Get data to clear Read/Write bits
U 122A, 0F00,003C,0180,0803,0000,122B  ;3121   VA_VA+4    ; Point to CNFG Reg B
U 122B, 0000,003C,0180,A800,0000,122C  ;3122  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 122C, 0818,0038,79B0,0800,0000,122D  ;3123  D_K[.30]    ; Prepare to form x30000780 to clear
        ;3124        ; ...CNFG Reg's C and D
U 122D, 0B00,003C,0180,0800,0000,122E  ;3125  D_D.SWAP    ; D = x30000000
        ;3126  Q_D,      ; Save in the Q Reg
U 122E, 0818,0038,CDE0,0800,0000,122F  ;3127   D_K[.F0]   ; D = xF0
U 122F, 0000,003C,0D80,0800,0084,7230  ;3128  SC_K[.3]    ; Must shift D Reg left 3 bits to get
        ;3129        ; ...x780
```

```
U 1230, 0D00,003C,0180,0800,0000,1231  ;3130  D_DAL.SC   ; D = x780
        ;3131  D_D.OR.Q,    ; D = x30000780
U 1231, 081D,0030,0180,0803,0000,1232  ;3132   VA_VA+4   ; Point to CNFG Reg C
U 1232, 000D,003C,0180,A800,0000,1233  ;3133  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 1233, 000D,003C,0180,0800,0000,1234  ;3134  NOP
U 1234, 000D,003C,0180,A800,0000,1235  ;3135  CACHE.P_D[LONG]   ; Do second write to make sure that
        ;3136       ; ...uSequencer Parity error bit
        ;3137       ; ...is clear
U 1235, 000D,003C,0180,0803,0000,1236  ;3138  VA_VA+4    ; Point to CNFG Reg D
U 1236, 000D,003C,0180,A800,0000,1237  ;3139  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 1237, 000D,003C,0180,0800,0000,1238  ;3140  NOP
        ;3141  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
        ;3142       ; ...Parity error bit
U 1238, 0000,003E,0180,A800,0000,0001  ;3143   RETURN1   ; Return to caller
        ;3144
        ;3145
```

```
            ;3146  .PAGE
            ;3147  .TOC "   Wait Loop Routine"
            ;3148 ;**
            ;3149 ; FUNCTIONAL DESCRIPTION:
            ;3150 ;
            ;3151 ; This routine is used to wait the specified number of machine cycles.
            ;3152 ; This routine is typically called to wait for an SBI write to
            ;3153 ; I/O space to complete.
            ;3154 ;
            ;3155 ; CALLING CONSTRAINT:
            ;3156 ;
            ;3157 ; =0
            ;3158 ;
            ;3159 ; INPUTS:
            ;3160 ;
            ;3161 ; d - Contains number of cycles to wait
            ;3162 ; alu.z condition code must not be set
            ;3163 ;
            ;3164 ; SIDE EFFECTS:
            ;3165 ;
            ;3166 ; d is destroyed
            ;3167 ;--
            ;3168 WAIT.LOOP:
U 1239, 0018,0038,6180,0800,0010,10A0  ;3169  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;3170      ; ...is not set
            ;3171 =0
            ;3172 W.LOOP:
            ;3173  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 10A0, 0819,0100,0580,0800,0010,10A0  ;3174       j/W.LOOP
U 10A1, 0000,003E,0180,0800,0000,0001  ;3175  return1    ; exit
            ;3176
            ;3177
```

```
;3178  .PAGE
;3179  .TOC "   DW780 INITIALIZE ROUTINE"
;3180 ;*************************************************************
;3181 ;**
;3182 ;
;3183 ; Functional Description:
;3184 ; ----------------------
;3185 ;
;3186 ;     Before  the  UBA  can  be  accessed  by  any  of  the  above
;3187 ; subroutines,  it should be initialized.  The following subroutine
;3188 ; will perform this task by  setting  bit  1  in  the  UBA  Control
;3189 ; Register.   Since the subroutine will be accessing one of the UBA
;3190 ; registers,  as  for  the  previous  subroutines  the  following
;3191 ; registers should be set up by the FIND.MS780E.AND.UBA subroutine:
;3192 ;
;3193 ;  ID[24] - Address of UBA Configuration Register
;3194 ;
;3195 ;  ID[25] - Address of Map Register 0
;3196 ;
;3197 ;  ID[26] - UBA Address Space (20100000)
;3198 ;
;3199 ;  ID[28] - Address of Data Path Register 1
;3200 ;
;3201 ;
;3202 ; Calling Constraint: =0
;3203 ; ------------------
;3204 ;
;3205 ;
;3206 ; Inputs:
;3207 ; ------
;3208 ;
;3209 ; See above description.
;3210 ;
;3211 ;
;3212 ; Outputs:
;3213 ; -------
;3214 ;
;3215 ; No explicit outputs. DW780 Initialized.
;3216 ;
;3217 ;
;3218 ; Side Effects:
;3219 ; ------------
;3220 ;
;3221 ; The Contents of the following registers will be lost:
;3222 ;
;3223 ; D, Q, VA
;3224 ;
;3225 ;--
;3226 ;*************************************************************
;3227 DW780.INIT:
U 123A, 0000,003C,91F0,2C00,0000,123B  ;3228  Q_ID[24]   ; Get Address of UBA CSR
U 123B, 0001,203C,0180,0800,0200,123C  ;3229  VA_ALU,ALU_Q   ; Move it to the VA
                ;3230  D_K[.1],   ; Get bit to force Init in UBA
U 123C, 0818,0038,0580,0803,0000,123D  ;3231   VA_VA+4   ; Point VA to UBA CONFIG Reg
U 123D, 0000,003C,0180,A800,0000,10A2  ;3232  CACHE.P_D[LONG]   ; Write the CONIFG Reg
```

```
            ;3233       ; ...forcing an INIT
U 10A2, 0000,003D,0180,0800,0000,123E  ;3234 =0 CALL[DW780.WAIT]  ; Wait for the DW780 to complete the
            ;3235       ; Initialize
U 10A3, 0000,003E,0180,0800,0000,0001  ;3236  RETURN1     ; Return to caller
            ;3237
            ;3238
```

```
            ;3239  .PAGE
            ;3240  .TOC "   DW780 WAIT INIT ROUTINE"
            ;3241 ;*************************************************************
            ;3242 ;**
            ;3243 ;
            ;3244 ; Functional Description:
            ;3245 ; ----------------------
            ;3246 ;
            ;3247 ;     This subroutine will test bit 16 of  the  UBA  Configuration
            ;3248 ; Register  and  it  will wait in a loop until this bit is set.  It
            ;3249 ; makes use of the same ID registers as  the  previously  described
            ;3250 ; subroutine (See DW780.INIT).
            ;3251 ;
            ;3252 ;
            ;3253 ;
            ;3254 ; Calling Constraint: =0
            ;3255 ; ------------------
            ;3256 ;
            ;3257 ;
            ;3258 ; Inputs:
            ;3259 ; ------
            ;3260 ;
            ;3261 ; ID Registers 24, 25, 26, and 28 must be setup as described
            ;3262 ; in the DW780.INIT routine.
            ;3263 ;
            ;3264 ; Outputs:
            ;3265 ; -------
            ;3266 ;
            ;3267 ; No explicit outputs.  RETURN1 when DW780 Init is complete.
            ;3268 ;
            ;3269 ;
            ;3270 ; Side Effects:
            ;3271 ; ------------
            ;3272 ;
            ;3273 ; The contents of the following registers will be lost:
            ;3274 ;
            ;3275 ; D, VA, Q, SC
            ;3276 ;
            ;3277 ;--
            ;3278 ;*************************************************************
            ;3279 DW780.WAIT:
U 123E, 0000,003C,91F0,2C00,0000,123F  ;3280  Q_ID[24]   ; Get UBA CSR Address
U 123F, 0001,203C,0180,0800,0200,1240  ;3281  VA_ALU,ALU_Q   ; Point the VA to UBA CSR
            ;3282 DW780.WAIT.LOOP:
            ;3283  D[LONG]_CACHE.P,   ; Read the contents of the UBA CSR
U 1240, 0000,003C,6580,C800,0084,7241  ;3284   SC_K[.10]  ; Point to bit d16
            ;3285  ALU_D.ANDNOT.MASK,   ; Is bit 16 set in the UBA CSR?
U 1241, 0001,0024,0180,0800,0010,1242  ;3286   CLK.UBCC  ; ...
U 1242, 0000,013C,0180,0800,0000,10A4  ;3287  Z?     ; ...
U 10A3, 0000,003E,0180,0800,0000,0001  ;3288 =0 RETURN1    ; OK. UBA Init finished
U 10A5, 0000,003C,0180,0800,0000,1240  ;3289  J/DW780.WAIT.LOOP  ; Repeat loop Init not finished yet
            ;3290
            ;3291
```

```
;3292  .PAGE
;3293  .TOC "   DW780 EXTNDED MASKED WRITE"
;3294 ;*************************************************************
;3295 ;**
;3296 ;
;3297 ; Functional Description:
;3298 ; ----------------------
;3299 ;
;3300 ;      This subroutine is mainly used  to  "force"  the  memory  to
;3301 ; perform an Extended Masked Write operation.  A DW780 will have to
;3302 ; be used for this operation since the 11/780 CPU does not have the
;3303 ; ability  to  execute such a function.  Before this subroutine is
;3304 ; entered it is assumed that:  the calling routine  has  configured
;3305 ; the  memory  controllers as desired, Register D contains the byte
;3306 ; offset and that  the  following  data  has  been  set  up  in  ID
;3307 ; registers by the FIND.MS780E.AND.UBA routine:
;3308 ;
;3309 ;
;3310 ;  ID[24] - Address of UBA Configuration Register
;3311 ;
;3312 ;  ID[25] - Address of Map Register 0
;3313 ;
;3314 ;  ID[26] - UBA Address Space (20100000)
;3315 ;
;3316 ;  ID[28] - Address of Data Path Register 1
;3317 ;
;3318 ;      The subroutine will set up Map  Register  0  with  the  page
;3319 ; address  of  the memory (Note:  The UBA will be programed so that
;3320 ; write operations will be done only to location 0 of  the  memory.
;3321 ; This  entails  that  the starting address of the memory should be
;3322 ; set to 0 before this subroutine is entered,  and  that  a  memory
;3323 ; array   board   should  always  be  present  in  slot  0.   These
;3324 ; restrictions are imposed in order to keep the tests simple.).
;3325 ; A write will be done to one of the eight bytes of a quad-word  in
;3326 ; the  UBA  Address  space followed by a purge of the Buffered Data
;3327 ; Path.  The purge will actually  force  the  UBA  to  execute  the
;3328 ; Extended Masked Write.
;3329 ;
;3330 ;      Due to the way in  which  the  UBA  works,  Extended  Masked
;3331 ; Writes  will  not  always be executed.  When addressing a byte in
;3332 ; the lower longword of a quad-word location  in  the  UBA  address
;3333 ; space and then purging the UBA, a Longword Masked Write will take
;3334 ; place.   Extended  Masked  Writes  will  only  take  place  when
;3335 ; addressing  a  byte in the upper longword of a quad-word location
;3336 ; in the UBA Address space.
;3337 ;
;3338 ;
;3339 ; Calling Constraint: =0
;3340 ; ------------------
;3341 ;
;3342 ;
;3343 ; Inputs:
;3344 ; ------
;3345 ;
;3346 ; ID[24] - UBA CSR Address
```

```
            ;3347 ; ID[25] - UBA Address of MR0
            ;3348 ; ID[26] - UBA Address space (20100000)
            ;3349 ; ID[28] - UBA Address of DPR1
            ;3350 ;
            ;3351 ;      D - Byte offset
            ;3352 ;
            ;3353 ; Outputs:
            ;3354 ; -------
            ;3355 ;
            ;3356 ; No explicit outputs.
            ;3357 ;
            ;3358 ;
            ;3359 ; Side Effects:
            ;3360 ; ------------
            ;3361 ;
            ;3362 ; The contents of the following registers will be lost:
            ;3363 ;
            ;3364 ; RC[6], D, Q, SC, VA,
            ;3365 ;
            ;3366 ;--
            ;3367 ;*************************************************************
            ;3368 =0
            ;3369 DW780.WRITE:
            ;3370   RC[6]_D,    ; Save Byte offset
U 10A6, 0001.003D.0180.09B0.0000.123A  ;3371    CALL[DW780.INIT] ; Initialize DW780
U 10A7, 0818.0038.4180.0800.0000.1243  ;3372  D_K[.80]    ; Form x80200000 in D Reg
            ;3373   D_D.SWAP,    ; ...D = x80000000
U 1243, 0B00.003C.2180.0800.0084.7244  ;3374    SC_K[.14]  ; ...Point to bit d21
            ;3375   SC_SC+1,    ; ...SC = d21
U 1244, 0000.003C.95F0.2C00.0080.D245  ;3376    Q_ID[25]  ; Get Address of MR0
U 1245, 0801.001C.0180.0800.0000.1246  ;3377  D_D.ORNOT.MASK    ; Finally D = B0200000
U 1246, 0001.203C.0180.0800.0200.1247  ;3378  VA_ALU,ALU_Q    ; Point VA to MR0
U 1247, 0000.003C.99F0.2C00.0000.1248  ;3379  Q_ID[26]    ; Get UBA Address Space (x20100000)
U 1248, 0000.003C.0180.A800.0000.1249  ;3380  CACHE.P_D[LONG]    ; Set up UBA MR0
            ;3381   VA_ALU,ALU_Q.OR.RC[6],   ; Get UBA Address OR Byte Offset
U 1249, 0F11.2030.0180.0930.0200.124A  ;3382    D_0    ; Data to write to UBA byte address
U 124A, 0000.803C.0180.A800.0000.124B  ;3383  CACHE.P_D[BYTE]    ; Force a Byte Write to UBA Address
            ;3384          ; ...space
U 124B, 0000.003C.A1F0.2C00.0000.124C  ;3385  Q_ID[28]    ; Get UBA Address of DPR1
U 124C, 0001.203C.0180.0800.0200.124D  ;3386  VA_ALU,ALU_Q    ; Point VA to DPR1
U 124D, 0818.0038.4180.0800.0000.124E  ;3387  D_K[.80]    ; Get x80000000 in D Reg
U 124E, 0B00.003C.0180.0800.0000.124F  ;3388  D_D.SWAP    ; ...
U 124F, 0000.003C.0180.A800.0000.10A8  ;3389  CACHE.P_D[LONG]    ; Purge the UBA Data Path forcing
            ;3390          ; ...the Extended Masked Write
            ;3391 =0 D_K[.10],    ; Wait now for UBA Cycle to complete
U 10A8, 0818.0039.6580.0800.0000.1239  ;3392    CALL[WAIT.LOOP]  ; ...
U 10A9, 0000.003E.0180.0800.0000.0001  ;3393  RETURN1    ; Return to caller
            ;3394
            ;3395
```

```
;3396  .PAGE
;3397  .TOC "   DW780 LONGWORD READ ROUTINE"
;3398 ;*************************************************************
;3399 ;**
;3400 ;
;3401 ; Functional Description:
;3402 ; ----------------------
;3403 ;
;3404 ;
;3405 ;      This subroutine is used to force the  memory  to  execute  a
;3406 ; Longword  Read  Masked.    A  UBA is necessary for this operation
;3407 ; since the CPU can only execute Extended  Reads.   As  before  the
;3408 ; following  registers  should be set up by the FIND.MS780E.AND.UBA
;3409 ; routine:
;3410 ;
;3411 ;  ID[24] - Address of UBA Configuration Register
;3412 ;
;3413 ;  ID[25] - Address of Map Register 0
;3414 ;
;3415 ;  ID[26] - UBA Address Space (20100000)
;3416 ;
;3417 ;  ID[28] - Address of Data Path Register 1
;3418 ;
;3419 ;      When the subroutine is entered, Register D should hold a two
;3420 ; bit code for the type of operation to be performed.  These codes,
;3421 ; aside from designating the type of operation, will be used  as  a
;3422 ; Byte  Offset when the Longword Read will take place.  They are as
;3423 ; follows:
;3424 ;
;3425 ;  b00 - Read Word 0
;3426 ;
;3427 ;  b01 - Read Byte 1
;3428 ;
;3429 ;  b10 - Read Word 1
;3430 ;
;3431 ;  b11 - Read Byte 3
;3432 ;
;3433 ; The subroutine will set up Map Register 0 with the  Page  Address
;3434 ; of  the  memory  (See note for DW780.EXTND.WRITE.MSKD subroutine.
;3435 ; Same  restrictions  are  valid  here.).   Upon  decoding  of  the
;3436 ; information in Register D, appropriate action will be taken.
;3437 ;
;3438 ;
;3439 ; Calling Constraint: =0
;3440 ; ------------------
;3441 ;
;3442 ;
;3443 ; Inputs:
;3444 ; ------
;3445 ;
;3446 ; ID Registers 24,25,26,28 must be set up as detailed above.
;3447 ;
;3448 ; D Reg MUST hold the Byte Offset as described above
;3449 ;
;3450 ; Outputs:
```

```
            ;3451 ; -------
            ;3452 ;
            ;3453 ; Data Read from the memory will be returned in the D register
            ;3454 ;
            ;3455 ;
            ;3456 ; Side Effects:
            ;3457 ; ------------
            ;3458 ;
            ;3459 ; The contents of the following registers will be lost:
            ;3460 ;
            ;3461 ; RC[6], D, Q, SC, VA
            ;3462 ;
            ;3463 ;--
            ;3464 ;*****************************************************************
            ;3465 =0
            ;3466 DW780.READ:
            ;3467   RC[6]_D.AND.K[.3],   ; Save Byte Offset
U 10AA, 0019.0035.0D80.09B0.0000.123A   ;3468    CALL[DW780.INIT] ; Initialize the UBA
U 10AB, 0003.003C.0180.0AF8.0000.1250   ;3469   R[0F]_0     ; Clear all Traps
U 1250, 0818.0038.4180.0800.0000.1251   ;3470   D_K[.80]    ; Load a x80000000 in the D Register
U 1251, 0B00.003C.0180.0800.0000.1252   ;3471   D_D.SWAP    ; ...
U 1252, 0000.003C.95F0.2C00.0000.1253   ;3472   Q_ID[25]    ; Get UBA Address of MR0
U 1253, 0001.203C.0180.0800.0200.1254   ;3473   VA_ALU,ALU_Q    ; Point the VA to MR0
U 1254, 0000.003C.0180.A800.0000.10AC   ;3474   CACHE.P_D[LONG]    ; Set MR0
U 10AC, 0000.003D.8980.0800.0084.71C9   ;3475 =0 SET.TRAP.MASK[.D]   ; Enable Time-Out uTraps
U 10AD, 0000.003C.99F0.2C00.0000.1255   ;3476   Q_ID[26]    ; Get UBA Address space
            ;3477   VA_ALU,ALU_Q.OR.RC[6],   ; Point the VA to the UBA address space
U 1255, 0811.2030.0180.0930.0200.1256   ;3478    D_ALU    ; ...and the byte to be read
U 1256, 0000.193C.0180.0800.0000.103C   ;3479   D1-0?     ; See which case we have
            ;3480 =1100 D[WORD]_CACHE.P,   ; CASE 00: Read WORD 0
U 103C, 0000.403E.0180.C800.0000.0001   ;3481    RETURN1    ; ...and return to caller
            ;3482   D[BYTE]_CACHE.P,   ; CASE 01: Read BYTE 1
U 103D, 0000.803E.0180.C800.0000.0001   ;3483    RETURN1    ; ...and return to caller
            ;3484   D[WORD]_CACHE.P,   ; CASE 10: Read WORD 1
U 103E, 0000.403E.0180.C800.0000.0001   ;3485    RETURN1    ; ...and return to caller
            ;3486   D[BYTE]_CACHE.P,   ; CASE 11: Read BYTE 3
U 103F, 0000.803E.0180.C800.0000.0001   ;3487    RETURN1    ; ...and return to caller
            ;3488
            ;3489
```

```
;3490  .PAGE "TEST 1FF EXTENDED MASK WRITES"
;3491 ;*************************************************************
;3492 ;+++
;3493 ;
;3494 ; Functional Description:
;3495 ; ----------------------
;3496 ;
;3497 ; This test programs the DW780 to perfrom Extended Masked Writes to
;3498 ; memory location 0.  Each byte within the quadword will be accessed in
;3499 ; a loop.  The test starts by initializing quadword locaton 0 to all
;3500 ; ones.  Next, a byte of all zeroes is written to the DW780 Buffered
;3501 ; Data path.  Purging the DW780 will cause it to perfrom an Extended
;3502 ; Masked Write.  Location 0 is then read to verify the integrity of the
;3503 ; data.
;3504 ;
;3505 ;
;3506 ; Logic Description:
;3507 ; -----------------
;3508 ;
;3509 ;   N/A
;3510 ;
;3511 ; Error Logout:
;3512 ; ------------
;3513 ;
;3514 ; See individual error reports.
;3515 ;
;3516 ; Sync Point Description:
;3517 ; ----------------------
;3518 ;   N/A
;3519 ;
;3520 ;=============================================================
;3521 ;
;3522 ; Register Map:
;3523 ; ------------
;3524 ;
;3525 ;      RA,RB      Description:
;3526 ;      -----      -----------
;3527 ;
;3528 ; Not Used
;3529 ;
;3530 ;       RC        Description:
;3531 ;       --        -----------
;3532 ;
;3533 ; E I/O Base Address of MS780-E Currently Under Test
;3534 ;
;3535 ; D I/O Address of Configuration Register C/D
;3536 ;
;3537 ; C Expected Upper Memory Data
;3538 ;
;3539 ; B Expected Lower Memory Data
;3540 ;
;3541 ; A Received Upper Memory Data
;3542 ;
;3543 ; 9 Received Lower Memory Data
;3544 ;
```

```
            ;3545 ; 8 Byte Pointer/Loop Counter
            ;3546 ;
            ;3547 ; 7 Longword Pointer ( 0 = Lower , 1 = Upper )
            ;3548 ;
            ;3549 ;--
            ;3550 ;*************************************************************
            ;3551 =0
U 10AE, 0000.003D.0180.0800.0000.117F   ;3552 T.48: NEWTST   ; Signify the start of a new test
U 10AF, 0000.003C.0180.0800.0000.1030   ;3553  NOP
U 1030, 0000.003D.0180.0800.0000.107C   ;3554 =00 CALL[FIND.MS780E.AND.UBA]
            ;3555      ; Search out all MS780-E's and UBA's on the SBI
U 1031, 0000.003C.0180.0800.0000.127C   ;3556  J/T.48.EXIT  ; Either/or no MS780-E's/UBA's to test, exit
U 1032, 0000.003C.0180.0800.0000.1034   ;3557  NOP
            ;3558 =
            ;3559  ;
            ;3560  ; This is the restart point for the test when it finishes testing the
            ;3561  ; controller currently under test and needs to configure the next one
            ;3562  ; for testing.  If there is no more controllers to configure then the
            ;3563  ; test is exited.
            ;3564  ;
            ;3565 =00
            ;3566 RESTART.TEST.48:
U 1034, 0000.003D.01B0.0800.0000.106E   ;3567  CALL[START.TEST] ; Configure the controllers
U 1035, 0000.003C.01B0.0800.0000.127C   ;3568  J/T.48.EXIT  ; No more MS780-E's to test, exit this test
U 1036, 0003.003C.01B0.09B8.0000.1037   ;3569  RC[07]_0  ; Clear longword pointer (0=lower,1=upper)
U 1037, 0003.003C.01B0.09C0.0000.1257   ;3570  RC[08]_0  ; Set up a byte point and loop counter
U 1257, 001B.001C.19B0.09E0.0000.1258   ;3571  RC[0C]_NOT.K[ZERO] ; Load RC[0C] with expected upper data
U 1258, 001B.001C.49B0.09D8.0000.10B0   ;3572  RC[0B]_NOT.K[.FF] ; Load RC[0B] with expected lower data
            ;3573  ;
            ;3574  ; This is the loop point that will be entered 8 times to check 8
            ;3575  ; different bytes within a quadword.
            ;3576  ;
            ;3577 =0
            ;3578 T.48.INNER.LOOP.1:
U 10B0, 0000.003D.0180.0800.0000.1194   ;3579  ERLOOP   ; loop on error from here
U 10B1, 0000.003C.0180.0800.0000.10B2   ;3580  NOP
U 10B2, 0000.003D.0180.0800.0000.1078   ;3581 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10B3, 0018.0038.1980.0800.0200.1259   ;3582  VA_K[ZERO]  ; Point VA to location 0
U 1259, 081B.001C.1980.0800.0000.125A   ;3583  D_NOT.K[ZERO]  ; Load D with xFFFFFFFF to init location 0
U 125A, 0000.003C.0180.A800.0000.125B   ;3584  CACHE.P_D[LONG]  ; Initialize location 0 with xFFFFFFFF
U 125B, 0000.003C.0180.0803.0000.125C   ;3585  VA_VA+4   ; Update VA to init memory location 4
U 125C, 0000.003C.0180.A800.0000.125D   ;3586  CACHE.P_D[LONG]  ; Initialzie location 4 with xFFFFFFFF
U 125D, 0810.0038.0180.0940.0000.10B4   ;3587  D_RC[08]  ; Load D with the byte pointer
U 10B4, 0000.003D.0180.0800.0000.10A6   ;3588 =0 CALL[DW780.WRITE] ; Force the DW780 to perfrom the Extended
            ;3589      ; Masked Write
U 10B5, 0018.0038.1980.0800.0200.125E   ;3590  VA_K[ZERO]  ; Point VA to location 0
U 125E, 0000.003C.0180.C800.0000.125F   ;3591  D[LONG]_CACHE.P  ; Read Location 0 and
U 125F, 0001.003C.0180.09C8.0000.1260   ;3592  RC[09]_D  ; save in RC[09]
U 1260, 0000.003C.0180.0803.0000.1261   ;3593  VA_VA+4   ; Update the VA to read memory locatio 4
U 1261, 0000.003C.0180.C800.0000.1262   ;3594  D[LONG]_CACHE.P  ; Read location 4 and
U 1262, 0001.003C.0180.09D0.0000.1263   ;3595  RC[0A]_D  ; save in RC[0A]
U 1263, 0810.0038.0180.0948.0000.1264   ;3596  D_RC[09]  ; Move saved received data from location 0 to
            ;3597      ; the D register
            ;3598  ALU_D.XOR.RC[0B], ; Exclusive Or the received memory data with
U 1264, 0011.0020.0180.0958.0010.1265   ;3599  CLK.UBCC  ; the expected memory data
```

```
ZZ-ESKAR-V22  TEST 1FF EXTENDED MASK WRITES                                    SEQ 1975
ESKAR54.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:07          Page   9

U 1265, 0000,013C,0180,0800,0000,10B6  ;3600  Z?    ; Were they equal ?
U 10B6, 0000,003C,0180,0800,0000,10B8  ;3601 =0 J/T.48.BAD.DATA.LOWER ; No, call an error
U 10B7, 0000,003C,0180,0800,0000,10B9  ;3602  J/T.48.CHK.UPPER ; Yes, continue with the test
```

```
            ;3603  .PAGE
            ;3604 ;
            ;3605 ;////////////////////////////////////////////////////////////////////////////////
            ;3606 ;
            ;3607 ; Received memory data and Expected memory data for the lower longword did
            ;3608 ; not match.  Call an error.
            ;3609 ;
            ;3610 =0
            ;3611 T.48.BAD.DATA.LOWER:
U 10B8, 0000,003D,0180,0800,0084,705C  ;3612  ERROR2[.8]
            ;3613 ;
            ;3614 ;////////////////////////////////////////////////////////////////////////////////
            ;3615 ;
            ;3616 ; ERROR LOGOUT:
            ;3617 ;
            ;3618 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3619 ;  I/O Address of Configuration Register C/D
            ;3620 ;  Unused
            ;3621 ;  Expected Lower Memory Data
            ;3622 ;  Unused
            ;3623 ;  Received Lower Memory Data
            ;3624 ;  Byte Pointer/Loop Counter
            ;3625 ;  Longword Pointer ( 0 = Lower , 1 = Upper )
            ;3626 ;
            ;3627 ;////////////////////////////////////////////////////////////////////////////////
            ;3628 ;
            ;3629 T.48.CHK.UPPER:
U 10B9, 0810,0038,0180,0950,0000,1266  ;3630  D_RC[0A]  ; Load the D register with the saved received
            ;3631     ; memory data for the upper longword
            ;3632  ALU_D.XOR.RC[0C], ; Exclusive Or the received memory data with
U 1266, 0011,0020,0180,0960,0010,1267  ;3633  CLK.UBCC  ; the expected memory data
U 1267, 0000,013C,0180,0800,0000,10BA  ;3634  Z?   ; Were they equal ?
U 10BA, 0000,003C,0180,0800,0000,10BC  ;3635 =0 J/T.48.BAD.UPPER.DATA ; No, call an error
U 10BB, 0000,003C,0180,0800,0000,10BD  ;3636  J/T.48.UPDATE.TST.PAT ; Yes, continue with the test
```

```
            ;3637  .PAGE
            ;3638 ;
            ;3639 ;////////////////////////////////////////////////////////////////////////////////
            ;3640 ;
            ;3641 ; Received memory data and Expected memory data for the upper longword did
            ;3642 ; not match.  Call an error.
            ;3643 ;
            ;3644 =0
            ;3645 T.48.BAD.UPPER.DATA:
U 10BC, 0000,003D,0180,0800,0084,705C  ;3646  ERROR2[.8]
            ;3647 ;
            ;3648 ;////////////////////////////////////////////////////////////////////////////////
            ;3649 ;
            ;3650 ; ERROR LOGOUT:
            ;3651 ;
            ;3652 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3653 ;  I/O Address of Configuration Register C/D
            ;3654 ;  Expected Upper Memory Data
            ;3655 ;  Unused
            ;3656 ;  Received Upper Memory Data
            ;3657 ;  Unused
            ;3658 ;  Byte Pointer/Loop Counter
            ;3659 ;  Longword Pointer ( 0 = Lower , 1 = Upper )
            ;3660 ;
            ;3661 ;////////////////////////////////////////////////////////////////////////////////
            ;3662 ;
            ;3663  ;
            ;3664  ; Now update the Byte Pointer/Loop Counter by one and check to see
            ;3665  ; if it is equal to 8.  If it is then restart the test for the next
            ;3666  ; controller as all 8 bytes of the quadword have been tested.  If
            ;3667  ; not then check the 'transition' state.  This state is defined by
            ;3668  ; the Byte Pointer/Loop counter being equal to 4, which means that
            ;3669  ; the upper longword is going to be tested now and we need to shift
            ;3670  ; the upper and set the flag saying that we are shifting the upper.
            ;3671  ;
            ;3672 T.48.UPDATE.TST.PAT:
U 10BD, 0810,0038,0180,0940,0000,1268  ;3673  D_RC[08]  ; Load D with the Byte pointer/loop counter
U 1268, 0019,0014,0580,09C0,0000,1269  ;3674  RC[08]_D+K[.1]  ; Update Byte pointer/loop counter by 1
U 1269, 0810,0038,0180,0940,0000,126A  ;3675  D_RC[08]  ; Load D with the Byte pointer/loop counter
            ;3676  ALU_D.XOR.K[.8], ; Check Byte pointer/loop counter to see if
U 126A, 0019,0020,0180,0800,0010,126B  ;3677  CLK.UBCC  ; equal to 8, if yes then done with this
            ;3678      ; controller
U 126B, 0000,013C,0180,0800,0000,10BE  ;3679  Z?  ; Is it equal to 8?
U 10BE, 0000,003C,0180,0800,0000,126C  ;3680 =0 J/T.48.CHK.TRANSITION ; No, check for equal to 4 - transition state
U 10BF, 0000,003C,0180,0800,0000,1034  ;3681  J/RESTART.TEST.48 ; Yes, restart for the next controller
            ;3682  ;
            ;3683  ; Come here when the transition state is reached (RC[08] = 4).  Set
            ;3684  ; the longword flag saying that we are now working on the upper
            ;3685  ; upper longword.  Also, change expected lower longword value to
            ;3686  ; xFFFFFFFF and the upper to xFFFFFF00.
            ;3687  ;
            ;3688 T.48.CHK.TRANSITION:
U 126C, 0810,0038,0180,0940,0000,126D  ;3689  D_RC[08]  ; Load D back up with Byte pointer/loop counter
            ;3690  ALU_D.XOR.K[.4], ; Exclusive Or update byte point/loop counter
U 126D, 0019,0020,1180,0800,0010,126E  ;3691  CLK.UBCC  ; with x4.  When RC[08] becomes equal to 4
```

```
            ;3692     ; then shift upper longword
U 126E. 0000,013C,0180,0800,0000,10C0  ;3693  Z?    ; Is counter/pointer equal to 4 ?
U 10C0. 0000,003C,0180,0800,0000,1271  ;3694 =0 J/T.48.DO.SHIFT  ; No, perform the shift
U 10C1. 0018,0038,0580,09B8,0000,126F  ;3695  RC[07]_K[.1]  ; Set flag in RC[07] to specify working on the
            ;3696     ; upper longword
U 126F. 001B,001C,1980,09D8,0000,1270  ;3697  RC[0B]_NOT.K[ZERO] ; Load expected lower with xFFFFFFFF
            ;3698  RC[0C]_NOT.K[.FF], ; Load expected upper with xFFFFFF00
U 1270. 001B,001C,4980,09E0,0000,10B0  ;3699  J/T.48.INNER.LOOP.1 ; and restart for next pass
            ;3700  ;
            ;3701  ; Come here to shift the data by 8 bits for every byte that is being
            ;3702  ; tested.  Check the value of the longword flag (RC[07]) and if it
            ;3703  ; is equal to 0 then shift the lower longword.  If it is equal to 1
            ;3704  ; then shift the upper longword.
            ;3705  ;
            ;3706 T.48.DO.SHIFT:
U 1271. 0010,0038,0180,0938,0010,1272  ;3707  ALU_RC[07],CLK.UBCC ; Load ALU with longword flag
U 1272. 0000,013C,0180,0800,0000,10C2  ;3708  Z?    ; Is flag equal to zero ?
U 10C2. 0000,003C,0180,0800,0000,1277  ;3709 =0 J/T.48.SHIFT.UPPER ; No, shift the upper longword
U 10C3. 0010,0038,01C0,0960,0000,1273  ;3710  Q_RC[0C]  ; Yes, shift the lower, set-up Q for shifting
            ;3711     ; ones into the DAL
U 1273. 0000,003C,0180,0800,0084,7274  ;3712  SC_K[.8]  ; Load SC with x8 for shift left of 8
U 1274. 0810,0038,0180,0958,0000,1275  ;3713  D_RC[0B]  ; Load the D register with the lower longword
U 1275. 0D00,003C,0180,0800,0000,1276  ;3714  D_DAL.SC  ; Do the shift of 8 w/ones
            ;3715  RC[0B]_D,  ; Restore shifted value in RC[0B]
U 1276. 0001,003C,0180,09D8,0000,10B0  ;3716  J/T.48.INNER.LOOP.1 ; and do another loop
            ;3717 T.48.SHIFT.UPPER:
U 1277. 0010,0038,01C0,0958,0000,1278  ;3718  Q_RC[0B]  ; set-up Q for shifting ones into the DAL
U 1278. 0000,003C,0180,0800,0084,7279  ;3719  SC_K[.8]  ; Load SC with x8 for shift left of 8
U 1279. 0810,0038,0180,0960,0000,127A  ;3720  D_RC[0C]  ; Load the D register with the upper longword
U 127A. 0D00,003C,0180,0800,0000,127B  ;3721  D_DAL.SC  ; Do the shift of 8 w/ones
            ;3722  RC[0C]_D,  ; Restore shifted value in RC[0C]
U 127B. 0001,003C,0180,09E0,0000,10B0  ;3723  J/T.48.INNER.LOOP.1 ; and do another loop
            ;3724  ;
            ;3725  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;3726  ;
            ;3727 T.48.EXIT:
U 127C. 0000,003C,0180,0800,0000,10C4  ;3728  NOP
            ;3729
            ;3730
```

```
;3731   .PAGE "TEST 200 CRD/RDS ERRORS ON EXTENDED MASKED WRITE"
;3732  ;*************************************************************
;3733  ;***
;3734  ;
;3735  ; Functional Description:
;3736  ; ----------------------
;3737  ;
;3738  ; As in the previous test the DW780 is used to perfrom Extended Masked
;3739  ; Writes to Memory.  However, in this case CRD and RDS errors are
;3740  ; forced using Diagnostic Mode 2.
;3741  ;
;3742  ;   [Subtest 1:]
;3743  ;
;3744  ; This subtest initializes locations 0 and 4 on the array, and sets the
;3745  ; check bits in CNFG B register, so that a CRD error will be forced in
;3746  ; bit 0 when executing an Extended Masked Write.  Following the write
;3747  ; operation, a read will retreive the data for checking.  It is expected
;3748  ; that both the lower and the upper longowrds will have the error
;3749  ; corrected.
;3750  ;
;3751  ;   [Subtest 2:]
;3752  ;
;3753  ; This subtest initializes locations 0 and 4 on the array and sets the
;3754  ; check bits in CNFG B register so that a RDS error occurs when the DW780
;3755  ; performs an Extended Masked Write.  After the Write operation takes
;3756  ; place, the data on the array is checked and it is expected that it will
;3757  ; not be modified (ie., write cycle was aborted.)
;3758  ;
;3759  ;
;3760  ; Logic Description:
;3761  ; -----------------
;3762  ;
;3763  ;   N/A
;3764  ;
;3765  ; Error Logout:
;3766  ; ------------
;3767  ;
;3768  ; See individual error reports.
;3769  ;
;3770  ; Sync Point Description:
;3771  ; ----------------------
;3772  ;   N/A
;3773  ;
;3774  ;=============================================================
;3775  ;
;3776  ; Register Map:
;3777  ; ------------
;3778  ;
;3779  ;     RA,RB     Description:
;3780  ;     -----     -----------
;3781  ;
;3782  ; Not Used
;3783  ;
;3784  ;      RC       Description:
;3785  ;      --       -----------
```

```
            ;3786 ;
            ;3787 ;       E I/O Base Address of MS780-E Currently Under Test
            ;3788 ;
            ;3789 ; D I/O Address of Configuration Register C/D
            ;3790 ;
            ;3791 ; C Expected Upper Longword Data
            ;3792 ;
            ;3793 ; B Expected Lower Longword Data
            ;3794 ;
            ;3795 ; A Received Upper Longword Data
            ;3796 ;
            ;3797 ; 9 Received Lower Longword Data
            ;3798 ;
            ;3799 ;--
            ;3800 ;************************************************************
            ;3801 =0
U 10C4, 0000,003D,0180,0800,0000,117F  ;3802 T.49: NEWTST   ; Signify the start of a new test
U 10C5, 0000,003C,0180,0800,0000,1038  ;3803  NOP
U 1038, 0000,003D,0180,0800,0000,107C  ;3804 =00 CALL[FIND.MS780E.AND.UBA]
            ;3805      ; Search out all MS780-E's and UBA
U 1039, 0000,003C,0180,0800,0000,10F0  ;3806  J/T.49.EXIT  ; No MS780-E's to test or no UBA
U 103A, 0000,003C,0180,0800,0000,1040  ;3807  NOP
            ;3808 =
            ;3809  ;
            ;3810  ; This is the restart point for the test to exercise the next
            ;3811  ; controller on the system.  If there is no more controllers to
            ;3812  ; test then the test is exited.
            ;3813  ;
            ;3814 =00
            ;3815 RESTART.TEST.49:
U 1040, 0000,003D,0180,0800,0000,106E  ;3816  CALL[START.TEST] ; Configure the controllers
U 1041, 0000,003C,0180,0800,0000,10F0  ;3817  J/T.49.EXIT  ; No more controllers to configure, exit test
            ;3818 ;
            ;3819 ;================================================================================
            ;3820 ;
            ;3821 ; Subtest 1
            ;3822 ;
U 1042, 0000,003D,0180,0800,0000,119C  ;3823  SUBTEST   ; Update the subtest counter
U 1043, 0003,003C,0180,09E0,0000,127D  ;3824  RC[0C]_0  ; Expected data from upper longword
U 127D, 0003,003C,0180,09D8,0000,10C6  ;3825  RC[0B]_0  ; Expected data from lower longword
U 10C6, 0000,003D,0180,0800,0000,1194  ;3826 =0 ERLOOP   ; Loop on error from here
U 10C7, 0000,003C,0180,0800,0000,10C8  ;3827  NOP
U 10C8, 0000,003D,0180,0800,0000,1078  ;3828 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10C9, 0818,0038,0580,0800,0000,127E  ;3829  D_K[.1]   ; Data to be used to force the CRD error
U 127E, 0018,0038,1980,0800,0200,127F  ;3830  VA_K[ZERO]  ; Point VA to location 0
U 127F, 0000,003C,0180,A800,0000,1280  ;3831  CACHE.P_D[LONG]  ; Initialize location 0
U 1280, 0000,003C,0180,0803,0000,1281  ;3832  VA_VA+4   ; Point Va to location 4
U 1281, 0000,003C,0180,A800,0000,10CA  ;3833  CACHE.P_D[LONG]  ; Initialize location 4
U 10CA, 0818,0039,0980,0800,0000,11AF  ;3834 =0 SET.DIAG.MODE[.2] ; Set ECC Substitute Diagnostic Mode
U 10CB, 0000,003C,0180,0800,0000,10CC  ;3835  NOP
U 10CC, 0818,0039,8580,0800,0000,1223  ;3836 =0 SET.ECC[.C]  ; Set the ECC substitute bits for data of
            ;3837      ; all zeroes
U 10CD, 0818,0038,5D80,0800,0000,10CE  ;3838  D_K[.7]   ; Get byte pointer
U 10CE, 0000,003D,0180,0800,0000,10A6  ;3839 =0 CALL[DW780.WRITE] ; Force teh UBA to execute an Extended Masked
            ;3840      ; Write
```

```
U 10CF, 0000.003C.0180.0800.0000.10D0  ;3841  NOP
U 10D0, 0818.0039.0D80.0800.0000.11AF  ;3842 =0 SET.DIAG.MODE[.3] ; Set ECC bypass Diagnostic mode
U 10D1, 0018.0038.1980.0800.0200.1282  ;3843  VA_K[ZERO]  ; Point the VA to location 0
U 1282, 0000.003C.0180.C800.0000.1283  ;3844  D[LONG]_CACHE.P  ; Read test location 0 and
U 1283, 0001.003C.0180.09C8.0000.1284  ;3845  RC[09]_D  ; save in RC[09]
U 1284, 0000.003C.0180.0803.0000.1285  ;3846  VA_VA+4  ; Point the VA to location 4
U 1285, 0000.003C.0180.C800.0000.1286  ;3847  D[LONG]_CACHE.P  ; Read test location 4 and
U 1286, 0001.003C.0180.09D0.0000.1287  ;3848  RC[0A]_D  ; save in RC[0A]
U 1287, 0010.0038.0180.0948.0010.1288  ;3849  ALU_RC[09],CLK.UBCC ; Load ALU with lower received data
U 1288, 0000.013C.0180.0800.0000.10D2  ;3850  Z?  ; Is the data equal to zero ?
U 10D2, 0000.003C.0180.0800.0000.10D4  ;3851 =0 J/T.49.S1.BAD.LOWER ; No, call an error
U 10D3, 0000.003C.0180.0800.0000.10D5  ;3852  J/T.49.S1.GOOD.LOWER ; Yes, continue with the test
```

```
            ;3853  .PAGE
            ;3854 ;
            ;3855 ;////////////////////////////////////////////////////////////////////////////////
            ;3856 ;
            ;3857 ; Data received from the lower longword was bad.  Call an error.
            ;3858 ;
            ;3859 =0
            ;3860 T.49.S1.BAD.LOWER:
U 10D4, 0000,003D,5D80,0800,0084,705C  ;3861  ERROR2[.7]
            ;3862 ;
            ;3863 ;////////////////////////////////////////////////////////////////////////////////
            ;3864 ;
            ;3865 ; ERROR LOGOUT:
            ;3866 ;
            ;3867 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3868 ;  I/O Address of Configuration Register C/D
            ;3869 ;  Unused
            ;3870 ;  Expected Lower Longword Data
            ;3871 ;  Unused
            ;3872 ;  Received Lower Longword Data
            ;3873 ;  Unused
            ;3874 ;
            ;3875 ;////////////////////////////////////////////////////////////////////////////////
            ;3876 ;
            ;3877 T.49.S1.GOOD.LOWER:
U 10D5, 0010,0038,0180,0950,0010,1289  ;3878  ALU_RC[0A],CLK.UBCC ; Load the ALU with upper received data
U 1289, 0000,013C,0180,0800,0000,10D6  ;3879  Z?    ; Is the data equal to zero ?
U 10D6, 0000,003C,0180,0800,0000,10D8  ;3880 =0 J/T.49.S1.BAD.UPPER ; No, Call an error
U 10D7, 0000,003C,0180,0800,0000,10DA  ;3881  J/T.49.S2  ; Yes, continue with the test
```

```
                ;3882  .PAGE
                ;3883 ;
                ;3884 ;////////////////////////////////////////////////////////////////////////////////
                ;3885 ;
                ;3886 ; Data received from the upper longword was bad.  Call an error.
                ;3887 ;
                ;3888 =0
                ;3889 T.49.S1.BAD.UPPER:
U 10D8. 0000.003D.5D80.0800.0084.705C   ;3890  ERROR2[.7]
U 10D9. 0000.003C.0180.0800.0000.10DA   ;3891  NOP
                ;3892 ;
                ;3893 ;////////////////////////////////////////////////////////////////////////////////
                ;3894 ;
                ;3895 ; ERROR LOGOUT:
                ;3896 ;
                ;3897 ; DATA: I/O Base Address of MS780-E Currently Under Test
                ;3898 ;  I/O Address of Configuration Register C/D
                ;3899 ;  Expected Upper Longword Data
                ;3900 ;  Unused
                ;3901 ;  Received Upper Longword Data
                ;3902 ;  Unused
                ;3903 ;  Unused
                ;3904 ;
                ;3905 ;////////////////////////////////////////////////////////////////////////////////
                ;3906 ;
                ;3907 ;
                ;3908 ;================================================================================
                ;3909 ;
                ;3910 ; Subtest 2
                ;3911 ;
                ;3912 =0
                ;3913 T.49.S2:
U 10DA. 0000.003D.0180.0800.0000.119C   ;3914  SUBTEST   ; Update the Subtest counter
U 10DB. 001B.0038.0D80.09E0.0000.128A   ;3915  RC[0C]_K[.3]  ; Expected data from upper longword
U 128A. 001B.0038.0D80.09D8.0000.10DC   ;3916  RC[0B]_K[.3]  ; Expected data from lower longword
U 10DC. 0000.003D.0180.0800.0000.1194   ;3917 =0 ERLOOP   ; loop on error from here
U 10DD. 0000.003C.0180.0800.0000.10DE   ;3918  NOP
U 10DE. 0000.003D.0180.0800.0000.1078   ;3919 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10DF. 081B.0038.0D80.0800.0000.128B   ;3920  D_K[.3]   ; Data to be used to force the CRD error
U 128B. 001B.0038.1980.0800.0200.128C   ;3921  VA_K[ZERO]  ; Point VA to location 0
U 128C. 0000.003C.0180.A800.0000.128D   ;3922  CACHE.P_D[LONG]  ; Initialize location 0
U 128D. 0000.003C.0180.0803.0000.128E   ;3923  VA_VA+4   ; Point VA to location 4
U 128E. 0000.003C.0180.A800.0000.10E0   ;3924  CACHE.P_D[LONG]  ; Initialize location 4
U 10E0. 081B.0039.0980.0800.0000.11AF   ;3925 =0 SET.DIAG.MODE[.2] ; Set ECC substitute Diagnostic mode
U 10E1. 0000.003C.0180.0800.0000.10E2   ;3926  NOP
U 10E2. 081B.0039.8580.0800.0000.1223   ;3927 =0 SET.ECC[.C]  ; Set the ECC substitute bits for data of
                ;3928      ; all zeroes
U 10E3. 0818.0038.5D80.0800.0000.10E4   ;3929  D_K[.7]   ; Get byte pointer
U 10E4. 0000.003D.0180.0800.0000.10A6   ;3930 =0 CALL[DW780.WRITE] ; Force the UBA to execute an Extended
                ;3931      ; masked write
U 10E5. 0000.003C.0180.0800.0000.10E6   ;3932  NOP
U 10E6. 0818.0039.0D80.0800.0000.11AF   ;3933 =0 SET.DIAG.MODE[.3] ; Set Diagnostic mode to ECC bypass
U 10E7. 0018.0038.1980.0800.0200.128F   ;3934  VA_K[ZERO]  ; Point the VA to location 0
U 128F. 0000.003C.0180.C800.0000.1290   ;3935  D[LONG]_CACHE.P  ; Read test location 0 and
U 1290. 0001.003C.0180.09C8.0000.1291   ;3936  RC[09]_D  ; save in RC[09]
```

```
U 1291, 0000,003C,0180,0803,0000,1292  ;3937  VA_VA+4   ; Point the VA to location 4
U 1292, 0000,003C,0180,C800,0000,1293  ;3938  D[LONG]_CACHE.P  ; Read test location 4 and
U 1293, 0001,003C,0180,09D0,0000,1294  ;3939  RC[0A]_D  ; save in RC[0A]
U 1294, 0810,0038,0180,0948,0000,1295  ;3940  D_RC[09]  ; Load D with received data from location 0
          ;3941  ALU_D.XOR.RC[0B], ; Exclusive Or the received memory data with
U 1295, 0011,0020,0180,0958,0010,1296  ;3942  CLK.UBCC  ; the expected memory data
U 1296, 0000,013C,0180,0800,0000,10E8  ;3943  Z?   ; Were they equal ?
U 10E8, 0000,003C,0180,0800,0000,10EA  ;3944 =0 J/T.49.S2.BAD.LOWER ; No, call an error
U 10E9, 0000,003C,0180,0800,0000,10EB  ;3945  J/T.49.S2.GOOD.LOWER ; Yes, continue with the test
```

```
            ;3946  .PAGE
            ;3947 ;
            ;3948 ;////////////////////////////////////////////////////////////////////////////////
            ;3949 ;
            ;3950 ; Data received from the lower longword was bad.  Call an error.
            ;3951 ;
            ;3952 =0
            ;3953 T.49.S2.BAD.LOWER:
U 10EA, 0000,003D,5D80,0800,0084,705C  ;3954  ERROR2[.7]
            ;3955 ;
            ;3956 ;////////////////////////////////////////////////////////////////////////////////
            ;3957 ;
            ;3958 ; ERROR LOGOUT:
            ;3959 ;
            ;3960 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3961 ;  I/O Address of Configuration Register C/D
            ;3962 ;  Unused
            ;3963 ;  Expected Lower Longword Data
            ;3964 ;  Unused
            ;3965 ;  Received Lower Longword Data
            ;3966 ;  Unused
            ;3967 ;
            ;3968 ;////////////////////////////////////////////////////////////////////////////////
            ;3969 ;
            ;3970 T.49.S2.GOOD.LOWER:
U 10EB, 0810,0038,0180,0950,0000,1297  ;3971  D_RC[0A]  ; Load the D register with the upper expected
            ;3972  ALU_D.XOR.RC[0C], ; Exclusive Or the received memory data with
U 1297, 0011,0020,0180,0960,0010,1298  ;3973  CLK.UBCC  ; the expected memory data
U 1298, 0000,013C,0180,0800,0000,10EC  ;3974  Z?  ; Were they equal ?
U 10EC, 0000,003C,0180,0800,0000,10EE  ;3975 =0 J/T.49.S2.BAD.UPPER ; No, call an error
U 10ED, 0000,003C,0180,0800,0000,1040  ;3976  J/RESTART.TEST.49 ; Yes, restart this test for the next
            ;3977      ; controller
```

```
            ;3978  .PAGE
            ;3979 ;
            ;3980 ;////////////////////////////////////////////////////////////////////////////
            ;3981 ;
            ;3982 ; Data received from the upper longword was bad.  Call an error.
            ;3983 ;
            ;3984 =0
            ;3985 T.49.S2.BAD.UPPER:
U 10EE, 0000,003D,5D80,0800,0084,705C  ;3986  ERROR2[.7]
U 10EF, 0000,003C,0180,0800,0000,1040  ;3987  J/RESTART.TEST.49
            ;3988 ;
            ;3989 ;////////////////////////////////////////////////////////////////////////////
            ;3990 ;
            ;3991 ; ERROR LOGOUT:
            ;3992 ;
            ;3993 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;3994 ;  I/O Address of Configuration Register C/D
            ;3995 ;  Expected Upper Longword Data
            ;3996 ;  Unused
            ;3997 ;  Received Upper Longword Data
            ;3998 ;  Unused
            ;3999 ;  Unused
            ;4000 ;
            ;4001 ;////////////////////////////////////////////////////////////////////////////
            ;4002 ;
            ;4003  ;
            ;4004  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4005  ;
U 1299, 0000,003C,0180,0800,0000,10F0  ;4006  NOP
            ;4007 =0
            ;4008 T.49.EXIT:
U 10F0, 0000,003D,0180,0800,0000,1229  ;4009  CALL[MS780E.CLEANUP]   ; Clear MS780-E CNFG Regs
U 10F1, 0000,003C,0180,0800,0000,10F2  ;4010  NOP
            ;4011
            ;4012
```

```
;4013  .PAGE "TEST 201 LONGWORD READ"
;4014 ;*************************************************************
;4015 ;+++
;4016 ;
;4017 ; Functional Description:
;4018 ; ----------------------
;4019 ;
;4020 ; This sequence of tests makes use of the DW780 to perfrom longword reads
;4021 ; from memory.  Both subtests verify that the data returned is correct by
;4022 ; checking it against the data used to initialize the memory.
;4023 ;
;4024 ;   [Subtest 1:]
;4025 ;
;4026 ; This subtest initializes location 0 of the array to all ones and then
;4027 ; programs the DW780 to execute a longword read.
;4028 ;
;4029 ;   [Subtest 2:]
;4030 ;
;4031 ; Location 0 of the array is initialized to all ones.  The DW780 is then
;4032 ; programmed to execute a "Masked Read".
;4033 ;
;4034 ;
;4035 ; Logic Description:
;4036 ; -----------------
;4037 ;   N/A
;4038 ;
;4039 ; Error Logout:
;4040 ; ------------
;4041 ;
;4042 ; See individual error reports.
;4043 ;
;4044 ; Sync Point Description:
;4045 ; ----------------------
;4046 ;   N/A
;4047 ;
;4048 ;=============================================================
;4049 ;
;4050 ; Register Map:
;4051 ; ------------
;4052 ;
;4053 ;      RA,RB      Description:
;4054 ;      -----      -----------
;4055 ;
;4056 ; Not Used
;4057 ;
;4058 ;       RC        Description:
;4059 ;       --        -----------
;4060 ;
;4061 ; E I/O Base Address of MS780-E Currently Under Test
;4062 ;
;4063 ; D I/O Address of Configuration Register C/D
;4064 ;
;4065 ; C Expected Memory Data
;4066 ;
;4067 ; B Received Memory Data
```

```
            ;4068 ;
            ;4069 ; 8 Expected Micro Trap Flags
            ;4070 ;
            ;4071 ; 7 Received Micro Trap Flags
            ;4072 ;
            ;4073 ;--
            ;4074 ;*************************************************************
            ;4075 =0
U 10F2, 0000.003D.0180.0800.0000.117F  ;4076 T.50: NEWTST   ; Signify the start of a new test
U 10F3, 0000.003C.0180.0800.0000.1044  ;4077  NOP
U 1044, 0000.003D.0180.0800.0000.107C  ;4078 =00 CALL[FIND.MS780E.AND.UBA]
            ;4079      ; Search out all MS780-E's and a UBA
U 1045, 0000.003C.0180.0800.0000.1118  ;4080  J/T.50.EXIT  ; No MS780-E's to test or no UBA
U 1046, 0000.003C.0180.0800.0000.1048  ;4081  NOP
            ;4082 =
            ;4083  ;
            ;4084  ; This is the restart point for checking the next controller on the
            ;4085  ; system.  If there is no more controllers to test then the test
            ;4086  ; is exited.
            ;4087  ;
            ;4088 =00
            ;4089 RESTART.TEST.50:
U 1048, 0000.003D.0180.0800.0000.106E  ;4090  CALL[START.TEST] ; Configure the controllers
U 1049, 0000.003C.0180.0800.0000.1118  ;4091  J/T.50.EXIT  ; No more controllers to configure, exit test
            ;4092 ;
            ;4093 ;=============================================================================
            ;4094 ;
            ;4095 ; Subtest 1
            ;4096 ;
U 104A, 0000.003D.0180.0800.0000.119C  ;4097  SUBTEST   ; Update the subtest counter
U 104B, 0018.0038.1980.0800.0200.129A  ;4098  VA_K[ZERO]  ; Point VA to location 0
U 129A, 0818.0038.4980.0800.0000.129B  ;4099  D_K[.FF]  ; Load the D register with xFF for init of
            ;4100      ; location 0
U 129B, 0000.003C.0180.A800.0000.129C  ;4101  CACHE.P_D[LONG]  ; Init location 0
U 129C, 0000.003C.0180.0803.0000.129D  ;4102  VA_VA+4   ; Update the VA for location 4
U 129D, 0F00.003C.0180.0800.0000.129E  ;4103  D_0   ; Load the D register with 0 for initialize of
            ;4104      ; location 4
U 129E, 0000.003C.0180.A800.0000.129F  ;4105  CACHE.P_D[LONG]  ; Init Location 4
U 129F, 0818.0038.4980.0800.0000.12A0  ;4106  D_K[.FF]  ; Load the D register with xFF
U 12A0, 0B00.003C.0180.0800.0000.12A1  ;4107  D_D.SWAP  ; Swap D so it has xFF000000
U 12A1, 0001.003C.0180.09E0.0000.10F4  ;4108  RC[0C]_D  ; Load RC[0C] with xFF000000, this is the
            ;4109      ; expected memory data
U 10F4, 0000.003D.0180.0800.0000.1194  ;4110 =0 ERLOOP   ; loop on error from here
U 10F5, 0000.003C.0180.0800.0000.10F6  ;4111  NOP
U 10F6, 0000.003D.0180.0800.0000.1078  ;4112 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 10F7, 0818.0038.0580.0800.0000.10F8  ;4113  D_K[.1]   ; Load D with 1 to specify a byte read
U 10F8, 0000.003D.0180.0800.0000.10AA  ;4114 =0 CALL[DW780.READ] ; Do the byte Read through the UBA
U 10F9, 0001.003C.0180.09D8.0000.1050  ;4115  RC[0B]_D  ; Save the received data
U 1050, 0000.003D.0180.0800.0000.11C3  ;4116 =00 CHECK.UTRAP  ; Was there a time-out micro trap ?
U 1051, 0000.003C.0180.0800.0000.1053  ;4117  J/T.50.S1.NO.UTRAP ; No, continue with the test
```

```
            ;4118  .PAGE
            ;4119 ;
            ;4120 ;////////////////////////////////////////////////////////////////////////////////
            ;4121 ;
            ;4122 ; Got a time out micro trap on the dw780 read.  Call an error.
            ;4123 ;
U 1052. 0000,003D,0180,0800,0084,705C  ;4124  ERROR2[.8]
            ;4125 ;
            ;4126 ;////////////////////////////////////////////////////////////////////////////////
            ;4127 ;
            ;4128 ; ERROR LOGOUT:
            ;4129 ;
            ;4130 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4131 ;  I/O Address of Configuration Register C/D
            ;4132 ;  Unused
            ;4133 ;  Unused
            ;4134 ;  Unused
            ;4135 ;  Unused
            ;4136 ;  Expected Micro Trap Flags
            ;4137 ;  Received Micro Trap Flags
            ;4138 ;
            ;4139 ;////////////////////////////////////////////////////////////////////////////////
            ;4140 ;
            ;4141 T.50.S1.NO.UTRAP:
U 1053. 0810,0038,0180,0958,0000,12A2  ;4142  D_RC[0B]  ; Load the D register with the received data
            ;4143  ALU_D.XOR.RC[0C], ; Exclusive or the received memory data with
U 12A2. 0011,0020,0180,0960,0010,12A3  ;4144  CLK.UBCC  ; expected memory data
U 12A3. 0000,013C,0180,0800,0000,10FA  ;4145  Z?   ; Are they equal ?
U 10FA. 0000,003C,0180,0800,0000,10FC  ;4146 =0 J/T.50.S1.BAD.DATA ; No, call an error
U 10FB. 0000,003C,0180,0800,0000,10FE  ;4147  J/T.50.S2  ; Yes, continue with the test
```

```
            ;4148  .PAGE
            ;4149 ;
            ;4150 ;////////////////////////////////////////////////////////////////////////////
            ;4151 ;
            ;4152 ; Received memory data did not match Expected memory data.  Call an error.
            ;4153 ;
            ;4154 =0
            ;4155 T.50.S1.BAD.DATA:
U 10FC, 0000.003D.1180.0800.0084.705C  ;4156  ERROR2[.4]
U 10FD, 0000.003C.0180.0800.0000.10FE  ;4157  NOP
            ;4158 ;
            ;4159 ;////////////////////////////////////////////////////////////////////////////
            ;4160 ;
            ;4161 ; ERROR LOGOUT:
            ;4162 ;
            ;4163 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4164 ;  I/O Address of Configuration Register C/D
            ;4165 ;  Expected Memory Data
            ;4166 ;  Received Memory Data
            ;4167 ;
            ;4168 ;////////////////////////////////////////////////////////////////////////////
            ;4169 ;
            ;4170 ;
            ;4171 ;============================================================================
            ;4172 ;
            ;4173 ; Subtest 2
            ;4174 ;
            ;4175 =0
            ;4176 T.50.S2:
U 10FE, 0000.003D.0180.0800.0000.119C  ;4177  SUBTEST   ; Update the subtest counter
U 10FF, 0018.0038.1980.0800.0200.12A4  ;4178  VA_K[ZERO]  ; Point VA to location 0
U 12A4, 0818.0038.C180.0800.0000.12A5  ;4179  D_K[.FFFF]  ; Load the D register with xFFFF for init of
            ;4180      ; location 0
U 12A5, 0000.003C.0180.A800.0000.12A6  ;4181  CACHE.P_D[LONG]  ; Init location 0
U 12A6, 0000.003C.0180.0803.0000.12A7  ;4182  VA_VA+4   ; Update the VA for location 4
U 12A7, 0F00.003C.0180.0800.0000.12A8  ;4183  D_0   ; Load the D register with 0 for initialize of
            ;4184      ; location 4
U 12A8, 0000.003C.0180.A800.0000.12A9  ;4185  CACHE.P_D[LONG]  ; Init Location 4
U 12A9, 0018.0038.C180.09E0.0000.110A  ;4186  RC[0C]_K[.FFFF]  ; Load RC[0C] with Expected Memory Data
U 110A, 0000.003D.0180.0800.0000.1194  ;4187 =0 ERLOOP   ; loop on error from here
U 110B, 0000.003C.0180.0800.0000.1110  ;4188  NOP
U 1110, 0000.003D.0180.0800.0000.1078  ;4189 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1111, 0818.0038.1980.0800.0000.1112  ;4190  D_K[ZERO]  ; Load D with 0 to specify a byte read
U 1112, 0000.003D.0180.0800.0000.10AA  ;4191 =0 CALL[DW780.READ] ; Do the byte Read through the UBA
U 1113, 0001.003C.0180.09D8.0000.1054  ;4192  RC[0B]_D  ; Save the received data
U 1054, 0000.003D.0180.0800.0000.11C3  ;4193 =00 CHECK.UTRAP  ; Was there a time-out micro trap ?
U 1055, 0000.003C.0180.0800.0000.1057  ;4194  J/T.50.S2.NO.UTRAP ; No, continue with the test
```

```
          ;4195  .PAGE
          ;4196 ;
          ;4197 ;////////////////////////////////////////////////////////////////////////////////
          ;4198 ;
          ;4199 ; Got a time out micro trap on the dw780 read.  Call an error.
          ;4200 ;
U 1056, 0000,003D,0180,0800,0084,705C  ;4201  ERROR2[.8]
          ;4202 ;
          ;4203 ;////////////////////////////////////////////////////////////////////////////////
          ;4204 ;
          ;4205 ; ERROR LOGOUT:
          ;4206 ;
          ;4207 ; DATA: I/O Base Address of MS780-E Currently Under Test
          ;4208 ;  I/O Address of Configuration Register C/D
          ;4209 ;  Unused
          ;4210 ;  Unused
          ;4211 ;  Unused
          ;4212 ;  Unused
          ;4213 ;  Expected Micro Trap Flags
          ;4214 ;  Received Micro Trap Flags
          ;4215 ;
          ;4216 ;////////////////////////////////////////////////////////////////////////////////
          ;4217 ;
          ;4218 T.50.S2.NO.UTRAP:
U 1057, 0810,0038,0180,0958,0000,12AB  ;4219  D_RC[0B]  ; Load the D register with the received data
          ;4220  ALU_D.XOR.RC[0C], ; Exclusive or the received memory data with
U 12AB, 0011,0020,0180,0960,0010,12AC  ;4221  CLK.UBCC  ; expected memory data
U 12AC, 0000,013C,0180,0800,0000,1114  ;4222  Z?   ; Are they equal ?
U 1114, 0000,003C,0180,0800,0000,1116  ;4223 =0 J/T.50.S2.BAD.DATA ; No, call an error
U 1115, 0000,003C,0180,0800,0000,1048  ;4224  J/RESTART.TEST.50 ; Yes, restart this test to check the next
          ;4225     ; controller
```

```
            ;4226  .PAGE
            ;4227 ;
            ;4228 ;////////////////////////////////////////////////////////////////////////////
            ;4229 ;
            ;4230 ; Received memory data did not match Expected memory data.  Call an error.
            ;4231 ;
            ;4232 =0
            ;4233 T.50.S2.BAD.DATA:
U 1116, 0000.003D.1180.0800.0084.705C  ;4234  ERROR2[.4]
U 1117, 0000.003C.0180.0800.0000.1048  ;4235  J/RESTART.TEST.50
            ;4236 ;
            ;4237 ;////////////////////////////////////////////////////////////////////////////
            ;4238 ;
            ;4239 ; ERROR LOGOUT:
            ;4240 ;
            ;4241 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4242 ;  I/O Address of Configuration Register C/D
            ;4243 ;  Expected Memory Data
            ;4244 ;  Received Memory Data
            ;4245 ;
            ;4246 ;////////////////////////////////////////////////////////////////////////////
            ;4247 ;
            ;4248  ;
            ;4249  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4250  ;
            ;4251 =0
            ;4252 T.50.EXIT:
U 1118, 0000.003D.0180.0800.0000.1229  ;4253  CALL[MS780E.CLEANUP]    ; CLEAR THE MS780-E REGISTERS
U 1119, 0000.003C.0180.0800.0000.111A  ;4254  NOP
            ;4255
```

```
;4256  .PAGE "TEST 202 CRD/RDS ERRORS ON LONGWORD READS"
;4257 ;*************************************************************
;4258 ;+++
;4259 ;
;4260 ; Functional Description:
;4261 ; ---------------------
;4262 ;
;4263 ; The following tests make use of the DW780 to perfrom longword reads of
;4264 ; the memory array while forcing CRD/RDS errors.
;4265 ;
;4266 ;   [Subtest 1:]
;4267 ;
;4268 ; This subtest initializes locations 0 and 4 of the array to all zeroes.
;4269 ; By making use of diagnostic mode 2 bits <6:0> of CNFG B register are
;4270 ; set up so that a CRD error will be forced when a read operation takes
;4271 ; place.  The test passes if:  CRD bit set in the status register of the
;4272 ; DW780, CRD bit set in CNFG C/D register, the data returned was
;4273 ; corrected on the array and location 4 on the array remains unchanged
;4274 ; following the read.
;4275 ;
;4276 ;   [Subtest 2:]
;4277 ;
;4278 ; This subtest initializes locations 0 and 4 of the memory array to all
;4279 ; zeroes.  Using diagnostic mode 2 to force a RDS error when a read of
;4280 ; the array is attempted.  The test is successfull only if:  The RDS bit
;4281 ; is set in the status register of the DW780, RDS bit set in CNFG C/D
;4282 ; register and neither of locations 0 or 4 was modified in any way.
;4283 ;
;4284 ;
;4285 ; Logic Description:
;4286 ; ------------------
;4287 ;
;4288 ;   N/A
;4289 ;
;4290 ; Error Logout:
;4291 ; ------------
;4292 ;
;4293 ; See individual error reports.
;4294 ;
;4295 ; Sync Point Description:
;4296 ; ----------------------
;4297 ;   N/A
;4298 ;
;4299 ;=============================================================
;4300 ;
;4301 ; Register Map:
;4302 ; ------------
;4303 ;
;4304 ;     RA,RB      Description:
;4305 ;     -----      ------------
;4306 ;
;4307 ;       0        Masking Value (x30000600)
;4308 ;
;4309 ;
;4310 ;      RC        Description:
```

```
            ;4311 ;       --        -----------
            ;4312 ;
            ;4313 ; E I/O Base Address of MS780-E Currently Under Test
            ;4314 ;
            ;4315 ; D I/O Address of Configuration Register C/D
            ;4316 ;
            ;4317 ; C Expected Test Data
            ;4318 ;
            ;4319 ; B Received Test Data
            ;4320 ;
            ;4321 ; A Received Register Data
            ;4322 ;
            ;4323 ; 9 Received Memory Data
            ;4324 ;
            ;4325 ; 8 Expected Micro Trap Flags
            ;4326 ;
            ;4327 ; 7 Received Micro Trap Flags
            ;4328 ;
            ;4329 ;--
            ;4330 ;*************************************************************
            ;4331 =0
U 111A, 0000,003D,0180,0800,0000,117F  ;4332 T.51: NEWTST   ; Signify the start of a newtest
U 111B, 0000,003C,0180,0800,0000,1060  ;4333  NOP
U 1060, 0000,003D,0180,0800,0000,107C  ;4334 =00 CALL[FIND.MS780E.AND.UBA]
            ;4335      ; Search out all MS780-E's on the system and
            ;4336      ; a UBA
U 1061, 0000,003C,0180,0800,0000,1178  ;4337  J/T.51.EXIT  ; No MS780-E's on the SBI and/or no UBA
U 1062, 0000,003C,0180,0800,0000,1064  ;4338  NOP
            ;4339 =
            ;4340  ;
            ;4341  ; This is the test restart point for checking the next controller.
            ;4342  ; If there is no more controllers on the system then the test is
            ;4343  ; exited.
            ;4344  ;
            ;4345 =00
            ;4346 RESTART.TEST.51:
U 1064, 0000,003D,0180,0800,0000,106E  ;4347  CALL[START.TEST] ; Configure the Controllers
U 1065, 0000,003C,0180,0800,0000,1178  ;4348  J/T.51.EXIT  ; No more controllers to configure/test
U 1066, 0000,003C,0180,0800,0000,111C  ;4349  NOP
            ;4350 =
            ;4351 ;
            ;4352 ;=============================================================================
            ;4353 ;
            ;4354 ; Subtest 1
            ;4355 ;
            ;4356 =0
            ;4357 T.51.S1:
U 111C, 0000,003D,0180,0800,0000,119C  ;4358  SUBTEST   ; Update the Subtest counter
U 111D, 0000,003C,0180,0800,0000,111E  ;4359  NOP
U 111E, 0000,003D,0180,0800,0000,1194  ;4360 =0 ERLOOP   ; Loop on error from here
U 111F, 0000,003C,0180,0800,0000,1120  ;4361  NOP
U 1120, 0000,003D,0180,0800,0000,1078  ;4362 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1121, 0000,003C,0180,0800,0000,1122  ;4363  NOP
U 1122, 0000,003D,0180,0800,0000,1229  ;4364 =0 CALL[MS780E.CLEANUP] ; Cleanup the MS780-E Registers
U 1123, 0018,0038,1980,0800,0200,12AD  ;4365  VA_K[ZERO]  ; Point the VA to location 0
```

```
U 12AD, 0818,0038,0580,0800,0000,12AE  ;4366  D_K[.1]   ; Load the D register with Initialize Data
U 12AE, 0000,003C,0180,A800,0000,12AF  ;4367  CACHE.P_D[LONG]  ; Initialize test location 0
U 12AF, 0000,003C,0180,0803,0000,12B0  ;4368  VA_VA+4   ; Point the VA to location 4
U 12B0, 0000,003C,0180,A800,0000,1124  ;4369  CACHE.P_D[LONG]  ; Initialize test location 4
U 1124, 0818,0039,0980,0800,0000,11AF  ;4370 =0 SET.DIAG.MODE[.2] ; Set ECC substitute diagnostic mode
U 1125, 0000,003C,0180,0800,0000,1126  ;4371  NOP
U 1126, 0818,0039,8580,0800,0000,1223  ;4372 =0 SET.ECC[.C]  ; See the ECC substitute bits for data of
          ;4373      ; all zeroes
U 1127, 0F00,003C,0180,0800,0000,1128  ;4374  D_0   ; Signal a word read to the dw780.read routine
U 1128, 0000,003D,0180,0800,0000,10AA  ;4375 =0 CALL[DW780.READ] ; Execute the Word Read
U 1129, 0001,003C,0180,09C8,0000,1068  ;4376  RC[09]_D  ; Save the read data
U 1068, 0000,003D,0180,0800,0000,11C3  ;4377 =00 CHECK.UTRAP  ; Was there a time-out micro trap
U 1069, 0000,003C,0180,0800,0000,106B  ;4378  J/T.51.S1.NO.UTRAP ; No, continue with the test
```

```
            ;4379  .PAGE
            ;4380 ;
            ;4381 ;////////////////////////////////////////////////////////////////////////////////
            ;4382 ;
            ;4383 ; Got a time-out micro trap on the word read from memory.  Call an error.
            ;4384 ;
U 106A, 0000.003D.0180.0800.0084.705C  ;4385  ERROR2[.8]
            ;4386 ;
            ;4387 ;////////////////////////////////////////////////////////////////////////////////
            ;4388 ;
            ;4389 ; ERROR LOGOUT:
            ;4390 ;
            ;4391 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4392 ;  I/O Address of Configuration Register C/D
            ;4393 ;  Unused
            ;4394 ;  Unused
            ;4395 ;  Unused
            ;4396 ;  Unused
            ;4397 ;  Expected Micro Trap Flags
            ;4398 ;  Received Micro Trap Flags
            ;4399 ;
            ;4400 ;////////////////////////////////////////////////////////////////////////////////
            ;4401 ;
            ;4402 T.51.S1.NO.UTRAP:
U 106B, 0000.003C.E980.0800.0084.72B1  ;4403  SC_K[.24]  ; Get ready to read ID 24
U 12B1, 0000.003C.01F0.2400.0000.12B2  ;4404  Q_ID(SC)  ; Load Q with the UBA Base Address
U 12B2, 0019.2014.0180.0800.0200.12B3  ;4405  VA_Q+K[.8]  ; Point the VA to the UBA Status Register
U 12B3, 0000.003C.0180.C800.0000.12B4  ;4406  D[LONG]_CACHE.P  ; Read UBA status register and
U 12B4, 0001.003C.0180.09D0.0000.12B5  ;4407  RC[0A]_D  ; save in RC[0A]
U 12B5, 0000.003C.0180.0800.00B4.72B6  ;4408  SC_K[.8]  ; Get ready to mask bit 8
U 12B6, 0003.001C.0180.09E0.0000.12B7  ;4409  RC[0C]_NOT.MASK  ; Assert bit 8 for value of x100, expected data
            ;4410     ; in UBA status register
U 12B7, 0810.0038.0180.0950.0000.12B8  ;4411  D_RC[0A]  ; Load D with data from UBA status register
            ;4412  ALU_D.ANDNOT.MASK, ; AND received data with x100, this bit should
U 12B8, 0001.0024.0180.09D8.0000.12B9  ;4413  RC[0B]_ALU  ; be set
U 12B9, 0810.0038.0180.0958.0000.12BA  ;4414  D_RC[0B]  ; Load D with Status Register CRD bit
            ;4415  ALU_D.XOR.RC[0C], ; Exclusive Or the Received CRD bit with the
U 12BA, 0011.0020.0180.0960.0010.12BB  ;4416  CLK.UBCC  ; expected value
U 12BB, 0000.013C.0180.0800.0000.112A  ;4417  Z?  ; Was the CRD bit set in the UBA status reg. ?
U 112A, 0000.003C.0180.0800.0000.112C  ;4418 =0 J/T.51.S1.CRD.NOT.SET ; No, Call an error
U 112B, 0000.003C.0180.0800.0000.112D  ;4419  J/T.51.S1.CRD.SET ; Yes, continue with the test
```

```
            ;4420  .PAGE
            ;4421 ;
            ;4422 ;////////////////////////////////////////////////////////////////////////////////
            ;4423 ;
            ;4424 ; UBA Status Register bit 8 ( CRD Error ) was not set an should have been.
            ;4425 ; Call an error.
            ;4426 ;
            ;4427 =0
            ;4428 T.51.S1.CRD.NOT.SET:
U 112C, 0000,003D,5D80,0800,0084,705C  ;4429  ERROR2[.7]
            ;4430 ;
            ;4431 ;////////////////////////////////////////////////////////////////////////////////
            ;4432 ;
            ;4433 ; ERROR LOGOUT:
            ;4434 ;
            ;4435 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4436 ;  I/O Address of Configuration Register C/D
            ;4437 ;  Expected Bit 8 from UBA Status Register
            ;4438 ;  Received Bit 8 from UBA Status Regsiter
            ;4439 ;  Contents of UBA Status Register (not masked)
            ;4440 ;  Unused
            ;4441 ;  Unused
            ;4442 ;
            ;4443 ;////////////////////////////////////////////////////////////////////////////////
            ;4444 ;
            ;4445 T.51.S1.CRD.SET:
U 112D, 0003,003C,0180,09E0,0000,12BC  ;4446  RC[0C]_0  ; Load RC[0C] with expected memory data
U 12BC, 0810,0038,0180,0948,0000,12BD  ;4447  D_RC[09]  ; Load the D register with received data
U 12BD, 0001,003C,0180,09D8,0000,12BE  ;4448  RC[0B]_D  ; and save in RC[0B]
U 12BE, 0010,0038,0180,0958,0010,12BF  ;4449  ALU_RC[0B],CLK.UBCC ; Load ALU with received memory data
U 12BF, 0000,013C,0180,0800,0000,112E  ;4450  Z?    ; Is the received data equal to zero ?
U 112E, 0000,003C,0180,0800,0000,1130  ;4451 =0 J/T.51.S1.BAD.DATA.1 ; No, call an error
U 112F, 0000,003C,0180,0800,0000,1132  ;4452  J/T.51.S1.GOOD.DAT.1 ; Yes, continue with the test
```

```
            ;4453  .PAGE
            ;4454 ;
            ;4455 ;////////////////////////////////////////////////////////////////////////////////
            ;4456 ;
            ;4457 ; Data reveived was not equal to zero.  Call an error.
            ;4458 ;
            ;4459 =0
            ;4460 T.51.S1.BAD.DATA.1:
U 1130, 0000,003D,5D80,0800,0084,705C  ;4461  ERROR2[.7]
U 1131, 0000,003C,0180,0800,0000,1132  ;4462  NOP
            ;4463 ;
            ;4464 ;////////////////////////////////////////////////////////////////////////////////
            ;4465 ;
            ;4466 ; ERROR LOGOUT:
            ;4467 ;
            ;4468 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4469 ;  I/O Address of Configuration Register C/D
            ;4470 ;  Expected Memory Data
            ;4471 ;  Received Memory Data
            ;4472 ;  Contents of UBA Status Register
            ;4473 ;  Unused
            ;4474 ;  Unused
            ;4475 ;
            ;4476 ;////////////////////////////////////////////////////////////////////////////////
            ;4477 ;
            ;4478 =0
            ;4479 T.51.S1.GOOD.DAT.1:
U 1132, 0000,003D,0180,0800,0000,1194  ;4480  ERLOOP   ; Loop on error from here
U 1133, 0000,003C,0180,0800,0000,1134  ;4481  NOP
U 1134, 0000,003D,0180,0800,0000,1078  ;4482 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1135, 0000,003C,0180,0800,0000,1136  ;4483  NOP
U 1136, 0000,003D,0180,0800,0000,1229  ;4484 =0 CALL[MS780E.CLEANUP] ; Clean up the MS780 registers
U 1137, 0018,0038,1980,0800,0200,12C0  ;4485  VA_K[ZERO]  ; Point VA to location 0
U 12C0, 0818,0038,0580,0800,0000,12C1  ;4486  D_K[.1]   ; Load D with data to force CRD error
U 12C1, 0000,003C,0180,A800,0000,12C2  ;4487  CACHE.P_D[LONG]  ; Initialize location 0
U 12C2, 0000,003C,0180,0803,0000,12C3  ;4488  VA_VA+4   ; Point VA to location 4
U 12C3, 0000,003C,0180,A800,0000,1138  ;4489  CACHE.P_D[LONG]  ; Initialize location 4
U 1138, 0818,0039,0980,0800,0000,11AF  ;4490 =0 SET.DIAG.MODE[.2] ; Set ECC substitute diagnostic mode
U 1139, 0000,003C,0180,0800,0000,113A  ;4491  NOP
U 113A, 0818,0039,8580,0800,0000,1223  ;4492 =0 SET.ECC[.C]  ; Set the substitute EC bits for data of
            ;4493     ; all zeroes
U 113B, 0F00,003C,0180,0800,0000,113C  ;4494  D_0   ; Signal word read to dw780.read routine]
U 113C, 0000,003D,0180,0800,0000,10AA  ;4495 =0 CALL[DW780.READ] ; Execute the word read
U 113D, 001B,0010,ED80,0800,0082,12C4  ;4496  SC_K[.1B]+1  ; Get ready to mask bit 28 for value x10000000
U 12C4, 0803,001C,C180,0800,0000,12C5  ;4497  D_NOT.MASK  ; load D with value x10000000
U 12C5, 0000,003C,D980,0800,0084,72C6  ;4498  SC_K[.9]  ; Get ready to mask bit 9 for value x10000200
U 12C6, 0001,001C,0180,09E0,0000,12C7  ;4499  RC[0C]_D.ORNOT.MASK ; Load RC[0C] with x10000200 which is the
            ;4500     ; expected contents of CNFG Register C/D
U 12C7, 0010,0038,0180,0968,0200,12C8  ;4501  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 12C8, 0000,003C,0180,C800,0000,12C9  ;4502  D[LONG]_CACHE.P  ; Read Configuration Register C/D and
U 12C9, 0001,003C,0180,09D0,0000,12CA  ;4503  RC[0A]_D  ; save in RC[0A]
U 12CA, 0810,0038,0180,0960,0000,12CB  ;4504  D_RC[0C]  ; Load D with x10000200
U 12CB, 0500,003C,0180,0800,0000,12CC  ;4505  D_D.LEFT,SI/ZERO ; Shift D left one place with zero shifted in
            ;4506     ; to yield value of x20000400
U 12CC, 0010,0038,01C0,0960,0000,12CD  ;4507  Q_RC[0C]  ; Load Q with x10000200
```

```
U 12CD, 081D,0030,0180,0800,0000,12CE  ;4508  D_D.OR.Q  ; Or values x20000400 and x10000200 together to
          ;4509     ; yield value of x30000600, this value is used
          ;4510     ; to mask out the unwanted bits in CNFG Reg C/D
U 12CE, 0001,003C,0180,0A80,0000,12CF  ;4511  R[0]_D   ; Save mask value of x30000600 in R[0]
U 12CF, 0010,0038,01C0,0950,0000,12D0  ;4512  Q_RC[0A]  ; Load Q with received CNFG C/D data
U 12D0, 001D,0034,0180,09D8,0000,12D1  ;4513  RC[0B]_D.AND.Q  ; Mask out bits 29:28 & 10:9 from CNFG C/D
U 12D1, 0810,0038,0180,0958,0000,12D2  ;4514  D_RC[0B]  ; Load D with masked bits
          ;4515  ALU_D.XOR.RC[0C], ; Exclusive Or the expected bits with the
U 12D2, 0011,0020,0180,0960,0010,12D3  ;4516  CLK.UBCC  ; received bits
U 12D3, 0000,013C,0180,0800,0000,113E  ;4517  Z?   ; Were they equal ?
U 113E, 0000,003C,0180,0800,0000,1140  ;4518 =0 J/T.51.S1.BAD.CNFG.DATA ; No, Call an error
U 113F, 0000,003C,0180,0800,0000,1142  ;4519  J/T.51.S1.GOOD.CNFG.DATA; Yes, continue with the test
```

```
            ;4520  .PAGE
            ;4521 ;
            ;4522 ;////////////////////////////////////////////////////////////////////////////////
            ;4523 ;
            ;4524 ; Configuration Register bit 28 ( Error Log Request ) and bit 9 ( CRD Flag )
            ;4525 ; were not set.  Call an error.
            ;4526 ;
            ;4527 =0
            ;4528 T.51.S1.BAD.CNFG.DATA:
U 1140, 0000.003D.5D80.0800.0084.705C  ;4529  ERROR2[.7]
U 1141, 0000.003C.0180.0800.0000.1142  ;4530  NOP
            ;4531 ;
            ;4532 ;////////////////////////////////////////////////////////////////////////////////
            ;4533 ;
            ;4534 ; ERROR LOGOUT:
            ;4535 ;
            ;4536 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4537 ;  I/O Address of Configuration Register C/D
            ;4538 ;  Expected Bits 29:28 & 10:9 from Configuratin Register C/D
            ;4539 ;  Received Bits 29:28 & 10:9 from Configuratin Register C/D
            ;4540 ;  Contents of Configuration Register C/D (not masked)
            ;4541 ;  Unused
            ;4542 ;  Unused
            ;4543 ;
            ;4544 ;////////////////////////////////////////////////////////////////////////////////
            ;4545 ;
            ;4546 =0
            ;4547 T.51.S1.GOOD.CNFG.DATA:
U 1142, 0818.0039.0D80.0800.0000.11AF  ;4548  SET.DIAG.MODE[.3] ; Select ECC Bypass diagnostic mode
U 1143, 0003.003C.0180.09E0.0000.12D4  ;4549  RC[0C]_0  ; Load RC[0C] with expected data from memory
U 12D4, 0018.0038.1980.0800.0200.12D5  ;4550  VA_K[ZERO]  ; Point the VA to location 0
U 12D5, 0000.003C.0180.C800.0000.12D6  ;4551  D[LONG]_CACHE.P  ; Read test location 0 and
U 12D6, 0001.003C.0180.09D8.0000.12D7  ;4552  RC[0B]_D  ; save in RC[0B]
U 12D7, 0010.0038.0180.0958.0010.12D8  ;4553  ALU_RC[0B],CLK.UBCC ; Load the ALU with lower received data
U 12D8, 0000.013C.0180.0800.0000.1144  ;4554  Z?  ; Is it equal to zero ?
U 1144, 0000.003C.0180.0800.0000.1146  ;4555 =0 J/T.51.S1.BAD.DATA.2 ; No, call an error
U 1145, 0000.003C.0180.0800.0000.1147  ;4556  J/T.51.S1.GOOD.DATA.2 ; yes, continue with the test
```

```
            ;4557  .PAGE
            ;4558 ;
            ;4559 ;////////////////////////////////////////////////////////////////////////////
            ;4560 ;
            ;4561 ; Lower received data was not equal to zero.  Call an error.
            ;4562 ;
            ;4563 =0
            ;4564 T.51.S1.BAD.DATA.2:
U 1146, 0000,003D,5D80,0800,0084,705C  ;4565  ERROR2[.7]
            ;4566 ;
            ;4567 ;////////////////////////////////////////////////////////////////////////////
            ;4568 ;
            ;4569 ; ERROR LOGOUT:
            ;4570 ;
            ;4571 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4572 ;  I/O Address of Configuration Register C/D
            ;4573 ;  Expected Memory Data
            ;4574 ;  Received Memory Data
            ;4575 ;  Contents of Configuration Register C/D
            ;4576 ;  Unused
            ;4577 ;  Unused
            ;4578 ;
            ;4579 ;////////////////////////////////////////////////////////////////////////////
            ;4580 ;
            ;4581 T.51.S1.GOOD.DATA.2:
U 1147, 0018,0038,0580,09E0,0000,12D9  ;4582  RC[0C]_K[.1]  ; Load RC[0C] with expected data from memory
U 12D9, 0018,0038,1180,0800,0200,12DA  ;4583  VA_K[.4]  ; Point the VA to location 4
U 12DA, 0000,003C,0180,C800,0000,12DB  ;4584  D[LONG]_CACHE.P  ; Read test location 4 and
U 12DB, 0001,003C,0180,09D8,0000,12DC  ;4585  RC[0B]_D  ; save in RC[0B]
            ;4586  ALU_D.XOR.K[.1], ; Exlcusive Or the received upper data with
U 12DC, 0019,0020,0580,0800,0010,12DD  ;4587  CLK.UBCC  ; expected upper data
U 12DD, 0000,013C,0180,0800,0000,1148  ;4588  Z?  ; Are they equal ?
U 1148, 0000,003C,0180,0800,0000,114A  ;4589 =0 J/T.51.S1.BAD.DATA.3 ; No, call an error
U 1149, 0000,003C,0180,0800,0000,114C  ;4590  J/T.51.S2  ; yes, continue with the test
```

```
            ;4591   .PAGE
            ;4592 ;
            ;4593 ;////////////////////////////////////////////////////////////////////////////
            ;4594 ;
            ;4595 ; Upper received data was not equal to one.  Call an error.
            ;4596 ;
            ;4597 =0
            ;4598 T.51.S1.BAD.DATA.3:
U 114A, 0000.003D.5D80.0800.0084.705C   ;4599  ERROR2[.7]
U 114B, 0000.003C.0180.0800.0000.114C   ;4600  NOP
            ;4601 ;
            ;4602 ;////////////////////////////////////////////////////////////////////////////
            ;4603 ;
            ;4604 ; ERROR LOGOUT:
            ;4605 ;
            ;4606 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4607 ;  I/O Address of Configuration Register C/D
            ;4608 ;  Expected Memory Data
            ;4609 ;  Received Memory Data
            ;4610 ;  Contents of Configuration Regsiter C/D
            ;4611 ;  Unused
            ;4612 ;  Unused
            ;4613 ;
            ;4614 ;////////////////////////////////////////////////////////////////////////////
            ;4615 ;
            ;4616 ;
            ;4617 ;============================================================================
            ;4618 ;
            ;4619 ; Subtest 2
            ;4620 ;
            ;4621 =0
            ;4622 T.51.S2:
U 114C, 0000.003D.0180.0800.0000.119C   ;4623  SUBTEST   ; Update the Subtest counter
U 114D, 0000.003C.0180.0800.0000.114E   ;4624  NOP
U 114E, 0000.003D.0180.0800.0000.1194   ;4625 =0 ERLOOP   ; Loop on error from here
U 114F, 0000.003C.0180.0800.0000.1150   ;4626  NOP
U 1150, 0000.003D.0180.0800.0000.1078   ;4627 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1151, 0000.003C.0180.0800.0000.1152   ;4628  nop
U 1152, 0000.003D.0180.0800.0000.1229   ;4629 =0 CALL[MS780E.CLEANUP] ; Cleanup the MS780-E registers
U 1153, 0018.0038.1980.0800.0200.12DE   ;4630  VA_K[ZERO]  ; Point the VA to location 0
U 12DE, 0818.0038.0D80.0800.0000.12DF   ;4631  D_K[.3]   ; Load the D register with Initialize Data
U 12DF, 0000.003C.0180.A800.0000.12E0   ;4632  CACHE.P_D[LONG]  ; Initialize test location 0
U 12E0, 0000.003C.0180.0803.0000.12E1   ;4633  VA_VA+4   ; Point the VA to location 4
U 12E1, 0000.003C.0180.A800.0000.1154   ;4634  CACHE.P_D[LONG]  ; Initialize test location 4
U 1154, 0818.0039.098C.0800.0000.11AF   ;4635 =0 SET.DIAG.MODE[.2] ; Set ECC substitute diagnostic mode
U 1155, 0000.003C.0180.0800.0000.1156   ;4636  NOP
U 1156, 0818.0039.8580.0800.0000.1223   ;4637 =0 SET.ECC[.C]  ; See the ECC substitute bits for data of
            ;4638       ; all zeroes
U 1157, 0F00.003C.0180.0800.0000.1158   ;4639  D_0   ; Signal a word read to the dw780.read routine
U 1158, 0000.003D.0180.0800.0000.10AA   ;4640 =0 CALL[DW780.READ] ; Execute the Word Read
U 1159, 0001.003C.0180.09C8.0000.12E2   ;4641  RC[09]_D  ; Save the read data
U 12E2, 0000.003C.E980.0800.0084.72E3   ;4642  SC_K[.24]  ; Get ready to read ID 24
U 12E3, 0000.003C.01F0.2400.0000.12E4   ;4643  Q_ID(SC)  ; Load Q with the UBA Base Address
U 12E4, 0019.2014.0180.0800.0200.12E5   ;4644  VA_Q+K[.8]  ; Point the VA to the UBA Status Register
U 12E5, 0000.003C.0180.C800.0000.12E6   ;4645  D[LONG]_CACHE.P  ; Read UBA status register and
```

```
U 12E6, 0001.003C.0180.09D0.0000.12E7  ;4646  RC[0A]_D  ; save in RC[0A]
U 12E7, 0000.003C.D980.0800.0084.72E8  ;4647  SC_K[.9]  ; Get ready to mask bit 9
U 12E8, 0003.001C.0180.09E0.0000.12E9  ;4648  RC[0C]_NOT.MASK  ; Assert bit 9 for value of x200, expected data
        ;4649     ; in UBA status register
U 12E9, 0810.0038.0180.0950.0000.12EA  ;4650  D_RC[0A]  ; Load D with data from UBA status register
        ;4651  ALU_D.ANDNOT.MASK, ; AND received data with x100, this bit should
U 12EA, 0001.0024.0180.09D8.0000.12EB  ;4652  RC[0B]_ALU  ; be set
U 12EB, 0810.0038.0180.0958.0000.12EC  ;4653  D_RC[0B]  ; Load D with Status Register RDS bit
        ;4654  ALU_D.XOR.RC[0C], ; Exclusive Or the Received RDS bit with the
U 12EC, 0011.0020.0180.0960.0010.12ED  ;4655  CLK.UBCC  ; expected value
U 12ED, 0000.013C.0180.0800.0000.115A  ;4656  Z?  ; Was the RDS bit set in the UBA status reg. ?
U 115A, 0000.003C.0180.0800.0000.115C  ;4657 =0 J/T.51.S2.RDS.NOT.SET ; No, Call an error
U 115B, 0000.003C.0180.0800.0000.115E  ;4658  J/T.51.S2.RDS.SET ; Yes, continue with the test
```

```
            ;4659  .PAGE
            ;4660 ;
            ;4661 ;////////////////////////////////////////////////////////////////////////////
            ;4662 ;
            ;4663 ; UBA Status Register bit 9 ( RDS Error ) was not set an should have been.
            ;4664 ; Call an error.
            ;4665 ;
            ;4666 =0
            ;4667 T.51.S2.RDS.NOT.SET:
U 115C. 0000.003D,5D80.0800,0084.705C  ;4668  ERROR2[.7]
U 115D. 0000.003C,0180.0800,0000.115E  ;4669  NOP
            ;4670 ;
            ;4671 ;////////////////////////////////////////////////////////////////////////////
            ;4672 ;
            ;4673 ; ERROR LOGOUT:
            ;4674 ;
            ;4675 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4676 ;  I/O Address of Configuration Register C/D
            ;4677 ;  Expected Bit 9 from UBA Status Register
            ;4678 ;  Received Bit 9 from UBA Status Regsiter
            ;4679 ;  Contents of UBA Status Register (not masked)
            ;4680 ;  Unused
            ;4681 ;  Unused
            ;4682 ;
            ;4683 ;////////////////////////////////////////////////////////////////////////////
            ;4684 ;
            ;4685 =0
            ;4686 T.51.S2.RDS.SET:
U 115E. 0000.003D,0180.0800,0000.1194  ;4687  ERLOOP   ; Loop on error from here
U 115F. 0000.003C,0180.0800,0000.1160  ;4688  NOP
U 1160. 0000.003D,0180.0800,0000.1078  ;4689 =0 CALL[SBI.INIT]  ; Initialize the SBI
U 1161. 0000.003C,0180.0800,0000.1162  ;4690  NOP
U 1162. 0000.003D,0180.0800,0000.1229  ;4691 =0 CALL[MS780E.CLEANUP] ; Clean up the MS780 registers
U 1163. 0018.0038,1980.0800,0200.12EE  ;4692  VA_K[ZERO]  ; Point VA to location 0
U 12EE. 0818.0038,0D80.0800,0000.12EF  ;4693  D_K[.3]   ; Load D with data to force RDS error
U 12EF. 000D.003C,0180.A800,0000.12F0  ;4694  CACHE.P_D[LONG]  ; Initialize location 0
U 12F0. 000D.003C,0180.0803,0000.12F1  ;4695  VA_VA+4   ; Point VA to location 4
U 12F1. 000D.003C,0180.A800,0000.1164  ;4696  CACHE.P_D[LONG]  ; Initialize location 4
U 1164. 0818.0039,0980.0800,0000.11AF  ;4697 =0 SET.DIAG.MODE[.2] ; Set ECC substitute diagnostic mode
U 1165. 000D.003C,0180.0800,0000.1166  ;4698  NOP
U 1166. 0818.0039,8580.0800,0000.1223  ;4699 =0 SET.ECC[.C]  ; Set the substitute EC bits for data of
            ;4700     ; all zeroes
U 1167. 0F00.003C,0180.0800,0000.1168  ;4701  D_0   ; Signal word read to dw780.read routine]
U 1168. 0000.003D,0180.0800,0000.10AA  ;4702 =0 CALL[DW780.READ] ; Execute the word read
U 1169. 001B.0010,ED80.0800,0082.12F2  ;4703  SC_K[.1B]+1  ; Get ready to mask bit 28 for value x10000000
U 12F2. 0803.001C,0180.0800,0000.12F3  ;4704  D_NOT.MASK  ; load D with value x10000000
U 12F3. 0000.003C,F580.0800,0084.72F4  ;4705  SC_K[.A]  ; Get ready to mask bit 10 for value x10000400
U 12F4. 0001.001C,0180.09E0,0000.12F5  ;4706  RC[0C]_D.ORNOT.MASK ; Load RC[0C] with x10000400 which is the
            ;4707     ; expected contents of CNFG Register C/D
U 12F5. 0010.0038,0180.0968,0200.12F6  ;4708  VA_RC[0D]  ; Point VA to Configuration Register C/D
U 12F6. 0000.003C,0180.C800,0000.12F7  ;4709  D[LONG]_CACHE.P  ; Read Configuration Register C/D and
U 12F7. 0001.003C,0180.09D0,0000.12F8  ;4710  RC[0A]_D  ; save in RC[0A]
U 12F8. 0010.0038,01C0.0950,0000.12F9  ;4711  Q_RC[0A]  ; Load Q with received CNFG C/D data
U 12F9. 0800.003C,0180.0A00,0000.12FA  ;4712  D_R[0]   ; Load D with mask value of x30000600
U 12FA. 001D.0034,0180.09D8,0000.12FB  ;4713  RC[0B]_D.AND.Q  ; Mask out bits 29:28 & 10:9 from CNFG C/D
```

```
ZZ-ESKAR-V22  TEST 202 CRD/RDS ERRORS ON LONGWORD READS                                    SEQ 2005
ESKAR54.MCR                        MICRO2  1P(00)    26-JUL-85  17:08:07                 Page  12

U 12FB, 0810,0038,0180,0958,0000,12FC  ;4714  D_RC[0B]  ; Load D with masked bits
          ;4715  ALU_D.XOR.RC[0C], ; Exclusive Or the expected bits with the
U 12FC, 0011,0020,0180,0960,0010,12FD  ;4716  CLK.UBCC  ; received bits
U 12FD, 0000,013C,0180,0800,0000,116A  ;4717  Z?    ; Were they equal ?
U 116A, 0000,003C,0180,0800,0000,116C  ;4718 =0 J/T.51.S2.BAD.CNFG.DATA ; No, Call an error
U 116B, 0000,003C,0180,0800,0000,116E  ;4719  J/T.51.S2.GOOD.CNFG.DATA; Yes, continue with the test
```

```
            ;4720  .PAGE
            ;4721 ;
            ;4722 ;////////////////////////////////////////////////////////////////////////////////
            ;4723 ;
            ;4724 ; Configuration Register bit 28 ( Error Log Request ) and bit 10 ( RDS Flag )
            ;4725 ; were not set.  Call an error.
            ;4726 ;
            ;4727 =0
            ;4728 T.51.S2.BAD.CNFG.DATA:
U 116C, 0000.003D.5D80.0800.0084.705C  ;4729  ERROR2[.7]
U 116D, 0000.003C.0180.0800.0000.116E  ;4730  NOP
            ;4731 ;
            ;4732 ;////////////////////////////////////////////////////////////////////////////////
            ;4733 ;
            ;4734 ; ERROR LOGOUT:
            ;4735 ;
            ;4736 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4737 ;  I/O Address of Configuration Register C/D
            ;4738 ;  Expected Bits 29:28 & 10:9 from Configuration Register C/D
            ;4739 ;  Received Bits 29:28 & 10:9 from Configuration Register C/D
            ;4740 ;  Contents of Configuration Register C/D (not masked)
            ;4741 ;  Unused
            ;4742 ;  Unused
            ;4743 ;
            ;4744 ;////////////////////////////////////////////////////////////////////////////////
            ;4745 ;
            ;4746 =0
            ;4747 T.51.S2.GOOD.CNFG.DATA:
U 116E, 0818.0039.0D80.0800.0000.11AF  ;4748  SET.DIAG.MODE[.3] ; Select ECC Bypass diagnostic mode
U 116F, 0018.0038.0D80.09E0.0000.12FE  ;4749  RC[0C]_K[.3]  ; Load RC[0C] with expected data from memory
U 12FE, 0018.0038.1980.0800.0200.12FF  ;4750  VA_K[ZERO]  ; Point the VA to location 0
U 12FF, 0000.003C.0180.C800.0000.1300  ;4751  D[LONG]_CACHE.P  ; Read test location 0 and
U 1300, 0001.003C.0180.09D8.0000.1301  ;4752  RC[0B]_D  ; save in RC[0B]
            ;4753  ALU_D.XOR.K[.3], ; Exclusive Or received data with expected data
U 1301, 0019.0020.0D80.0800.0010.1302  ;4754  CLK.UBCC
U 1302, 0000.013C.0180.0800.0000.1170  ;4755  Z?  ; Were they equal ?
U 1170, 0000.003C.0180.0800.0000.1172  ;4756 =0 J/T.51.S2.BAD.DATA.2 ; No, call an error
U 1171, 0000.003C.0180.0800.0000.1173  ;4757  J/T.51.S2.GOOD.DATA.2 ; yes, continue with the test
```

```
            ;4758  .PAGE
            ;4759 ;
            ;4760 ;////////////////////////////////////////////////////////////////////////////
            ;4761 ;
            ;4762 ; Lower received data was not equal to 3.  Call an error.
            ;4763 ;
            ;4764 =0
            ;4765 T.51.S2.BAD.DATA.2:
U 1172. 0000.003D.5D80.0800.0084.705C  ;4766  ERROR2[.7]
            ;4767 ;
            ;4768 ;////////////////////////////////////////////////////////////////////////////
            ;4769 ;
            ;4770 ; ERROR LOGOUT:
            ;4771 ;
            ;4772 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4773 ;  I/O Address of Configuration Register C/D
            ;4774 ;  Expected Memory Data
            ;4775 ;  Received Memory Data
            ;4776 ;  Contents of Configuration Regsiter C/D (not used)
            ;4777 ;  Unused
            ;4778 ;  Unused
            ;4779 ;
            ;4780 ;////////////////////////////////////////////////////////////////////////////
            ;4781 ;
            ;4782 T.51.S2.GOOD.DATA.2:
U 1173. 0018.0038.0D80.09E0.0000.1303  ;4783  RC[0C]_K[.3]  ; Load RC[0C] with expected data from memory
U 1303. 0018.0038.1180.0800.0200.1304  ;4784  VA_K[.4]  ; Point the VA to location 4
U 1304. 0000.003C.0180.C800.0000.1305  ;4785  D[LONG]_CACHE.P  ; Read test location 4 and
U 1305. 0001.003C.0180.09D8.0000.1306  ;4786  RC[0B]_D  ; save in RC[0B]
            ;4787  ALU_D.XOR.K[.3], ; Exlcusive Or the received upper data with
U 1306. 0019.0020.0D80.0800.0010.1307  ;4788  CLK.UBCC  ; expected upper data
U 1307. 0000.013C.0180.0800.0000.1174  ;4789  Z?  ; Are they equal ?
U 1174. 0000.003C.0180.0800.0000.1176  ;4790 =0 J/T.51.S2.BAD.DATA.3 ; No, call an error
U 1175. 0000.003C.0180.0800.0000.1064  ;4791  J/RESTART.TEST.51 ; Yes, Restart this test to check the next
            ;4792     ; controller
```

```
            ;4793  .PAGE
            ;4794 ;
            ;4795 ;////////////////////////////////////////////////////////////////////////////////
            ;4796 ;
            ;4797 ; Upper received data was not equal to 3.  Call an error.
            ;4798 ;
            ;4799 =0
            ;4800 T.51.S2.BAD.DATA.3:
U 1176, 0000.003D.5D80.0800.0084.705C  ;4801  ERROR2[.7]
U 1177, 0000.003C.0180.0800.0000.1308  ;4802  NOP
            ;4803 ;
            ;4804 ;////////////////////////////////////////////////////////////////////////////////
            ;4805 ;
            ;4806 ; ERROR LOGOUT:
            ;4807 ;
            ;4808 ; DATA: I/O Base Address of MS780-E Currently Under Test
            ;4809 ;  I/O Address of Configuration Register C/D
            ;4810 ;  Expected Memory Data
            ;4811 ;  Received Memory Data
            ;4812 ;  Contents of Configuration Regsiter C/D (not masked)
            ;4813 ;  Unused
            ;4814 ;  Unused
            ;4815 ;
            ;4816 ;////////////////////////////////////////////////////////////////////////////////
            ;4817 ;
            ;4818  ;
            ;4819  ; This is where we exit the test.  ALL DONE!!!!!!!!!!!!!!!!!!!!!!!!!
            ;4820  ;
U 1308, 0000.003C.0180.0800.0000.1178  ;4821  NOP
            ;4822 =0
            ;4823 T.51.EXIT:
U 1178, 0000.003D.0180.0800.0000.1229  ;4824  CALL[MS780E.CLEANUP]    ; Clear Memory CNFG Regs
U 1179, 0000.003C.0180.0800.0000.117A  ;4825  NOP
            ;4826
            ;4827
```

```
            ;4828  .PAGE "TEST 203 RESERVED"
            ;4829 =0
U 117A, 0000,003D,0180,0800,0000,117F  ;4830 TXX1: NEWTST
U 117B, 0000,003C,0180,0800,0000,117C  ;4831  NOP
            ;4832
```

```
ZZ-ESKAR-V22  TEST 204 RESEREVED                                                SEQ 2010
ESKAR54.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:07           Page  13

            ;4833  .PAGE "TEST 204 RESEREVED"
            ;4834 =0
U 117C, 0000,003D,0180,0800,0000,117F  ;4835 TXX2: NEWTST
U 117D, 0000,003C,0180,0800,0000,1309  ;4836  NOP
            ;4837
```

```
            ;4838  .PAGE "DUMMY NEWTST"
U 1309. 0000,003D,0180,0800,0000,117F  ;4839 DUM: NEWTST
            ;4840
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2012
ESKAR54.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:07     Page  13
:                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2356= 2358= 2360= 2365=
U 1008  2379= 2381= 2382= 2386= 2345= 2346= 2347= 2348=
U 1010  2402= 2403= 2404= 2412= 2414= 2416= 2418= 2419=
U 1018  2804= 2805= 2806= 2807= 2808= 2809= 2810= 2811=
U 1020  2462= 2463= 2464= 2466= 1912= 1913= 1902 2844=
U 1028  2510= 2511= 2512= 2514= 1965= 1967= 1903 2845=
U 1030  3554= 3556= 3557= 1904 3567= 3568= 3569= 3570=
U 1038  3804= 3806= 3807= 1906 3481= 3483= 3485= 3487=
U 1040  3816= 3817= 3823= 3824= 4078= 4080= 4081= 1907
U 1048  4090= 4091= 4097= 4098= 1978= 1979= 2021: 1914
U 1050  4116= 4117= 4124= 4142= 4193= 4194= 4201= 4219=
U 1058  2039= 2040= 2067: 1915 2085= 2086= 2101: 1916
U 1060  4334= 4337= 4338= 1917 4347= 4348= 4349= 1918
U 1068  4377= 4378= 4385= 4403= 2208= 2209= 2334= 2335=
U 1070  2337= 2342= 2572= 2577= 2582= 2587= 2592= 2593=
U 1078  2637= 2638= 2715= 2716= 2782= 2783= 2792= 2793=
U 1080  2794= 2796= 2799= 2800= 2841= 2842= 2863= 2864=
U 1088  2876= 2877= 2878= 2879= 2886= 2887= 2893= 2895=
U 1090  2898= 2899= 2920= 2921= 2924= 2925= 2930= 2932=
U 1098  3022= 3023= 3026= 3027= 3038= 3039= 3042= 3043=
U 10A0  3174= 3175= 3234= 3236= 3288= 3289= 3371= 3372=
U 10A8  3392= 3393= 3468= 3469= 3475= 3476= 3552= 3553=
U 10B0  3579= 3580= 3581= 3582= 3588= 3590= 3601= 3602=
U 10B8  3612= 3630= 3635= 3636= 3646= 3673= 3680= 3681=
U 10C0  3694= 3695= 3709= 3710= 3802= 3803= 3826= 3827=
U 10C8  3828= 3829= 3834= 3835= 3836= 3838= 3839= 3841=
U 10D0  3842= 3843= 3851= 3852= 3861= 3878= 3880= 3881=
U 10D8  3890= 3891= 3914= 3915= 3917= 3918= 3919= 3920=
U 10E0  3925= 3926= 3927= 3929= 3930= 3932= 3933= 3934=
U 10E8  3944= 3945= 3954= 3971= 3975= 3976= 3986= 3987=
U 10F0  4009= 4010= 4076= 4077= 4110= 4111= 4112= 4113=
U 10F8  4114= 4115= 4146= 4147= 4156= 4157= 4177= 4178=
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895: 1919 4187= 4188= 1896: 1897: 1898: 1899:
U 1110  4189= 4190= 4191= 4192= 4223= 4224= 4234= 4235=
U 1118  4253= 4254= 4332= 4333= 4358= 4359= 4360= 4361=
U 1120  4362= 4363= 4364= 4365= 4370= 4371= 4372= 4374=
U 1128  4375= 4376= 4418= 4419= 4429= 4446= 4451= 4452=
U 1130  4461= 4462= 4480= 4481= 4482= 4483= 4484= 4485=
U 1138  4490= 4491= 4492= 4494= 4495= 4496= 4518= 4519=
U 1140  4529= 4530= 4548= 4549= 4555= 4556= 4565= 4582=
U 1148  4589= 4590= 4599= 4600= 4623= 4624= 4625= 4626=
U 1150  4627= 4628= 4629= 4630= 4635= 4636= 4637= 4639=
U 1158  4640= 4641= 4657= 4658= 4668= 4669= 4687= 4688=
U 1160  4689= 4690= 4691= 4692= 4697= 4698= 4699= 4701=
U 1168  4702= 4703= 4718= 4719= 4729= 4730= 4748= 4749=
U 1170  4756= 4757= 4766= 4783= 4790= 4791= 4801= 4802=
U 1178  4824= 4825= 4830= 4831= 4835= 4836= 1920 1954
U 1180  1955 1956 1957 1958 1959 1960 1961 1962
U 1188  1963 1964 1968 1969 1970 1971 1972 1973
U 1190  1974 1975 1976 1977 1996 2019 2020 2065
U 1198  2066 2149 2150 2151 2169 2171 2197 2198
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2199 2200 2201 2202 2203 2204 2206 2207
U 11A8  2343 2344 2369 2389 2390 2391 2398 2538
U 11B0  2539 2540 2541 2542 2543 2567 2615 2639
U 11B8  2640 2641 2642 2660 2661 2662 2663 2681
U 11C0  2682 2683 2684 2708 2709 2710 2711 2713
U 11C8  2714 2743 2744 2745 2784 2785 2786 2787
U 11D0  2788 2797 2798 2802 2803 2816 2817 2819
U 11D8  2820 2821 2823 2828 2829 2830 2832 2838
U 11E0  2839 2840 2843 2846 2848 2849 2850 2851
U 11E8  2852 2853 2854 2855 2856 2858 2860 2861
U 11F0  2866 2872 2874 2875 2883 2884 2888 2889
U 11F8  2890 2891 2892 2900 2901 2902 2904 2905
U 1200  2906 2907 2908 2909 2915 2916 2918 2919
U 1208  2922 2927 2928 2929 2955 2956 2957 2958
U 1210  2986 2987 2989 2990 2991 3018 3020 3021
U 1218  3025 3029 3030 3031 3032 3034 3035 3036
U 1220  3037 3040 3041 3067 3068 3069 3070 3071
U 1228  3072 3119 3121 3122 3123 3125 3127 3128
U 1230  3130 3132 3133 3134 3135 3138 3139 3140
U 1238  3143 3169 3228 3229 3231 3232 3280 3281
U 1240  3284 3286 3287 3374 3376 3377 3378 3379
U 1248  3380 3382 3383 3385 3386 3387 3388 3389
U 1250  3470 3471 3472 3473 3474 3478 3479 3571
U 1258  3572 3583 3584 3585 3586 3587 3591 3592
U 1260  3593 3594 3595 3596 3599 3600 3633 3634
U 1268  3674 3675 3677 3679 3689 3691 3693 3697
U 1270  3699 3707 3708 3712 3713 3714 3716 3718
U 1278  3719 3720 3721 3723 3728 3825 3830 3831
U 1280  3832 3833 3844 3845 3846 3847 3848 3849
U 1288  3850 3879 3916 3921 3922 3923 3924 3935
U 1290  3936 3937 3938 3939 3940 3942 3943 3973
U 1298  3974 4006 4099 4101 4102 4103 4105 4106
U 12A0  4107 4108 4144 4145 4179 4181 4182 4183
U 12A8  4185 4186 2127: 4221 4222 4366 4367 4368
U 12B0  4369 4404 4405 4406 4407 4408 4409 4411
U 12B8  4413 4414 4416 4417 4447 4448 4449 4450
U 12C0  4486 4487 4488 4489 4497 4498 4499 4501
U 12C8  4502 4503 4504 4505 4507 4508 4511 4512
U 12D0  4513 4514 4516 4517 4550 4551 4552 4553
U 12D8  4554 4583 4584 4585 4587 4588 4631 4632
U 12E0  4633 4634 4642 4643 4644 4645 4646 4647
U 12E8  4648 4650 4652 4653 4655 4656 4693 4694
U 12F0  4695 4696 4704 4705 4706 4708 4709 4710
U 12F8  4711 4712 4713 4714 4716 4717 4750 4751
U 1300  4752 4754 4755 4784 4785 4786 4788 4789
U 1308  4821 4839
U 1310 - 13EF Unused
U 13F0  2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8  2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2014
ESKAR54.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:07     Page  14

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR54        794

Used              794
Remaining

Total microwords used in memory U: 794
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                    SEQ 2015
ESKAR54.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:07   Page  14


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR54.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR55.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 2016

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1842 MICRO TRAP HANDLER
; 1933 Micro Monitor Interface Routines
; 1934   Newtest Routine
; 1991   ERROR1 WITH PARAMETER
; 2017   ERROR 2 WITH PARAMETER
; 2106   ASCII MESSAGES
; 2184   RESET SUBTEST NUMBER
; 2208   Set Trap Mask
; 2238   Disable Cache Routine
; 2287   Initialize the SBI
; 2384   CONFIGURE MS780-E/H
; 2459   Find MS780-E Memory
; 2604   Generate IO Address
; 2631   Set Starting Address
; 2665   ENABLE NEXT MS780-E
; 2717   Select Controller
; 2756   SELECT UPPER CONTROLLER
; 2805   SELECT LOWER CONTROLLER
; 2959 TEST 205 SBI INVALIDATE WITH GROUP 0 ADDRESS PARITY ERROR
; 3059 TEST 206 SBI INVALIDATE WITH GROUP 1 ADDRESS PARITY ERROR
; 3163 TEST 207 CPU READ AND SBI INVALIDATE SIMULTANEOUSLY
; 3571 TEST 208 SBI REQUESTS 4, 5, 6, OR 7
; 3927 TEST 209 RESERVED
; 3932 TEST 20A RESERVED
; 3937 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR55
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842   .TOC " MICRO TRAP HANDLER"
            ;1843 ;*
            ;1844 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1845 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
            ;1846 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1847 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1848 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1849 ;
            ;1850 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1851 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1852 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1853 ; R[0F] AND DO A RETURN0.
            ;1854 ;
            ;1855 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1856 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1857 ; TO THE CONSOLE.
            ;1858 ;
            ;1859 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
            ;1860 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
            ;1861 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
            ;1862 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
            ;1863 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
            ;1864 ;-
            ;1865
U 1100, 0000.003C.19C0.0A78.0084.7001   ;1866 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101, 0000.003C.05C0.0A78.0084.7001   ;1867 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000.003C.09C0.0A78.0084.7001   ;1868 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103, 0000.003C.0DC0.0A78.0084.7001   ;1869 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104, 0000.003C.11C0.0A78.0084.7001   ;1870 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105, 0000.003C.D5C0.0A78.0084.7001   ;1871 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106, 0000.003C.D9C0.0A78.0084.7001   ;1872 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000.003C.5DC0.0A78.0084.7001   ;1873 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108, 0000.003C.01C0.0A78.0084.7001   ;1874 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C, 0000.003C.85C0.0A78.0084.7001   ;1875 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D, 0000.003C.89C0.0A78.0084.7001   ;1876 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E, 0000.003C.65C0.0A78.0084.7001   ;1877 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F, 0000.003C.61C0.0A78.0084.702E   ;1878 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
            ;1879
U 1001, 0001.2024.0180.0800.0010.1004   ;1880 TRPH0: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1004, 0000.013C.0180.0800.0000.1002   ;1881   Z?     ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000   ;1882 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1883
            ;1884
            ;1885 ;*
            ;1886 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1887 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1888 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1889 ;
            ;1890 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1891 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1892 ;       FAULT STATUS REGISTER
            ;1893 ;       SBI ERROR REGISTER
            ;1894 ;       TIMEOUT ADDRESS REGISTER
            ;1895 ;       MAINTENANCE REGISTER
```

```
            ;1896 ;       CACHE PARITY REGISTER
            ;1897 ;       TB ERROR REGISTER 1
            ;1898 ;       TB ERROR REGISTER 0
            ;1899 ;-
            ;1900
U 1003, 0018.0038.1D80.09E8.0000.1024  ;1901 TRPH1: RC[0D]_SC
U 1024, 0000.003D.D980.0800.0084.717F  ;1902 =0 SETRCF[.9]
U 1025, 0000.003C.49F0.2C00.0000.1006  ;1903  Q_ID[TBER0]
U 1006, 0001.203C.4DF0.2DB0.0000.1013  ;1904  RC[6]_Q,Q_ID[TBER1]
U 1013, 0001.203C.65F0.2DB8.0000.1014  ;1905  RC[7]_Q,Q_ID[SBI.ERR]
U 1014, 0001.203C.6DF0.2DD8.0000.1016  ;1906  RC[0B]_Q,Q_ID[FAULT]
U 1016, 0001.203C.75F0.2DE0.0000.1023  ;1907  RC[0C]_Q,Q_ID[MAINT]
U 1023, 0001.203C.79F0.2DC8.0000.1026  ;1908  RC[9]_Q,Q_ID[PARITY]
U 1026, 0001.203C.69F0.2DC0.0000.102B  ;1909  RC[8]_Q,Q_ID[TIME.ADDR]
U 102B, 0001.203C.81F0.2DD0.0000.1000  ;1910  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1164  ;1911 1000: RC[0E]_Q,ERROR1
            ;1912
            ;1913 ;*
            ;1914 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
            ;1915 ;-
            ;1916
U 102E, 0001.2024.0180.0800.0010.104F  ;1917 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 104F, 0000.013C.0180.0800.0000.102C  ;1918  Z?     ; BRANCH IF NO
U 102C, 0001.2036.0180.0AF8.0000.0001  ;1919 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1  ; RETURN
U 102D, 0000.003C.0180.0800.0000.1003  ;1920  J/TRPH1     ; REPORT UNEXPECTED TRAP
            ;1921
            ;1922
            ;1923 ;*
            ;1924 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1925 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1926 ;  NEWTST
            ;1927 ;  ERLOOP
            ;1928 ;  ERROR1
            ;1929 ;  ERROR2
            ;1930 ;
            ;1931
```

```
            ;1932  .PAGE
            ;1933  .TOC " Micro Monitor Interface Routines"
            ;1934  .TOC "   Newtest Routine"
            ;1935 ;**
            ;1936 ; FUNCTIONAL DESCRIPTION:
            ;1937 ;
            ;1938 ; This routine Initializes the following registers:
            ;1939 ;  PSL    to 1F
            ;1940 ;  CES    to 0
            ;1941 ;  ACC.CS    to disable accellerator
            ;1942 ;  SIR    to 0
            ;1943 ;  CLK.CS    to stop interval timer
            ;1944 ;  TBER0    to disable the TB
            ;1945 ;  SBI.MAINT to     0
            ;1946 ;  SBI.ERR    to 0
            ;1947 ;  FAULT    to 0
            ;1948 ;  COMP    to 0
            ;1949 ;  RC[0E]    to  0
            ;1950 ;  R[0F]    to 0
            ;1951 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1952 ; A SCOPE call is then made to the Monitor.
            ;1953 ;
            ;1954 ; CALLING CONSTRAINT:
            ;1955 ;
            ;1956 ; =0
            ;1957 ;
            ;1958 ; SIDE EFFECTS:
            ;1959 ;
            ;1960 ; D Reg is destroyed
            ;1961 ; SC is destroyed
            ;1962 ;--
U 105B, 0000,003C,65F8,0800,0084,705F  ;1963 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 105F, 0818,0038,8D80,0800,0000,1109  ;1964  d_k[.1f]  ; D=1F
U 1109, 0D00,003C,0180,0800,0000,114F  ;1965  d_dai.sc  ; D=001F0000
U 114F, 0000,003C,3D80,3C00,0000,1150  ;1966  id[psl]_d  ; psl = 001F0000
U 1150, 0818,0038,0580,0800,0000,1151  ;1967  d_k[.1]  ; Clear the CES register
U 1151, 0D00,003C,0180,0800,0000,1152  ;1968  d_dai.sc  ; D = 00010000
U 1152, 0F00,003C,3180,3C00,0000,1153  ;1969  id[ces]_d,d_0  ; ces = 00010000
U 1153, 0000,003C,5D80,3C00,0000,1154  ;1970  id[acc.cs]_d  ; acc.cs = 0
U 1154, 0000,003C,3980,3C00,0000,1156  ;1971  id[sir]_d  ; sir = 0
U 1156, 0000,003C,2980,3C00,0000,1157  ;1972  id[clk.cs]_d  ; clk.cs = 0
U 1157, 0000,003C,4980,3C00,0000,103E  ;1973  id[tber0]_d  ; tber0 = 0
U 103E, 0000,003D,0180,0800,0000,1197  ;1974 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1975  intrpt.strobe,  ; Strobe interrupts
U 103F, 0818,0038,C180,0800,4000,1158  ;1976      d_k[.ffff]  ; Setup to clear SBI error register and
U 1158, 0801,0028,6580,3C00,0000,1159  ;1977  id[sbi.err]_d,d_not.d ; and fault register
U 1159, 0F00,003C,6D80,3C00,0000,115A  ;1978  id[fault]_d,d_0  ; ...
U 115A, 0000,003C,6D80,3C00,0000,115B  ;1979  id[fault]_d  ; fault = 0
U 115B, 0000,003C,7580,3C00,0000,115C  ;1980  id[maint]_d  ; MAINT = 0
U 115C, 0000,003C,6580,3C00,0000,115D  ;1981  id[sbi.err]_d  ; sbi error reg = 0
U 115D, 0000,003C,7180,3C00,0000,115E  ;1982  id[comp]_d  ; comp reg = 0
U 115E, 0000,003C,E580,3C00,0000,115F  ;1983  id[T9]_d  ; Clear code for subroutine START.TEST
U 115F, 0001,003C,0180,09F0,0000,1160  ;1984  rc[0e]_d  ;
U 1160, 0001,003C,0180,0AF8,0000,1161  ;1985  r[0f]_d  ; Clear expected trap mask
U 1161, 0001,003C,0180,09F8,0000,1162  ;1986  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 2023
ESKAR55.MCR                        MICRO2  1P(00)    26-JUL-85  17:09:04               Page    5

U 1162, 0F03,003C,0180,09E8,0000,105E  ;1987  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
        ;1988
U 1163, 0818,0038,0980,0800,0000,105E  ;1989 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
```

```
            ;1990  .PAGE
            ;1991  .TOC "    ERROR1 WITH PARAMETER"
            ;1992 ;**
            ;1993 ; FUNCTIONAL DESCRIPTION:
            ;1994 ;
            ;1995 ; This subroutine takes  as parameter the number of arguments
            ;1996 ; to be printed to the console in the case of an error logout.
            ;1997 ;
            ;1998 ; CALLING CONSTRAINT:
            ;1999 ;
            ;2000 ; =0
            ;2001 ;INPUTS:
            ;2002 ;
            ;2003 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2004 ;--
            ;2005 =0
            ;2006 ERR1.ARG:
U 1040, 0000,003D,0180,0800,0000,117F  ;2007  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1041, 0000,003C,0180,0800,0000,1164  ;2008  J/ERROR1  ; Go to ERROR1
            ;2009 ;
            ;2010 ;
            ;2011
U 1164, 0810,0038,0180,0978,0000,1165  ;2012 ERROR1: D_RC[0F]
U 1165, 0819,0034,4D80,0800,0000,104E  ;2013  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2014 104E: D_D.OR.K[.4],J/CALLCON
            ;2015
```

```
            ;2016  .PAGE
            ;2017  .TOC "    ERROR 2 WITH PARAMETER"
            ;2018 ;++
            ;2019 ; FUNCTIONAL DESCRIPTION:
            ;2020 ;
            ;2021 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2022 ; except that in this case after setting the number of arguments
            ;2023 ; too the console, control is transferred to routine ERROR2.
            ;2024 ;
            ;2025 ; INPUTS:
            ;2026 ;
            ;2027 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2028 ;
            ;2029 ;--
            ;2030 =0
            ;2031 ERR2.ARG:
U 1042. 0000.003D.0180.0800.0000.117F  ;2032  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1043. 0000.003C.0180.0800.0000.1166  ;2033  J/ERROR2  ; Go to ERROR2
            ;2034 ;
            ;2035 ;
            ;2036
U 1166. 0810.0038.0180.0978.0000.1167  ;2037 ERROR2: D_RC[0F]
U 1167. 0819.0034.4D80.0800.0000.105A  ;2038  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;2039 105A: D_D.OR.K[.6],J/CALLCON
            ;2040
            ;2041 105E:
            ;2042 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;2043  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2044
            ;2045 ;+
            ;2046 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2047 ;-
            ;2048
U 13F0. 0000.003C.0180.0800.0000.13F1  ;2049 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;2050 13F1: NOP
U 13F2. 0000.003C.0180.0800.0000.13F3  ;2051 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;2052 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;2053 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;2054 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;2055 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;2056 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;2057 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;2058 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;2059 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;2060 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;2061 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;2062 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;2063 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1168  ;2064 13FF: NOP
            ;2065
            ;2066
            ;2067 ;+
            ;2068 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2069 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2070 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
```

```
            ;2071 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2072 ;
            ;2073 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2074 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2075 ;
            ;2076 ;   =0 D_0,CALL,J/DC5
            ;2077 ;    NOP
            ;2078 ;   =0 CALL,J/DC5
            ;2079 ;-
            ;2080
U 1168. 0018.0038.1980.0800.0000.1178   ;2081 DC0: ALU_0(K),J/S00
U 1169. 0018.0038.1980.0800.0000.1179   ;2082 DC1: ALU_0(K),J/S01
U 116A. 0018.0038.1980.0800.0000.117A   ;2083 DC2: ALU_0(K),J/S10
U 116B. 0018.0038.1980.0800.0000.117B   ;2084 DC3: ALU_0(K),J/S11
U 116C. 0018.0038.0980.0800.0000.1178   ;2085 DC4: ALU_K[.2],J/S00
U 116D. 0018.0038.0980.0800.0000.1179   ;2086 DC5: ALU_K[.2],J/S01
U 116E. 0018.0038.09B0.0800.0000.117A   ;2087 DC6: ALU_K[.2],J/S10
U 116F. 0018.0038.0980.0800.0000.117B   ;2088 DC7: ALU_K[.2],J/S11
U 1170. 0018.0038.0580.0800.0000.1178   ;2089 DC8: ALU_K[.1],J/S00
U 1171. 0018.0038.0580.0800.0000.1179   ;2090 DC9: ALU_K[.1],J/S01
U 1172. 0018.0038.0580.0800.0000.117A   ;2091 DCA: ALU_K[.1],J/S10
U 1173. 0018.0038.0580.0800.0000.117B   ;2092 DCB: ALU_K[.1],J/S11
U 1174. 0018.0038.C180.0800.0000.1178   ;2093 DCC: ALU_K[.FFFF],J/S00
U 1175. 0018.0038.C180.0800.0000.1179   ;2094 DCD: ALU_K[.FFFF],J/S01
U 1176. 0018.0038.C180.0800.0000.117A   ;2095 DCE: ALU_K[.FFFF],J/S10
U 1177. 0018.0038.C180.0800.0000.117B   ;2096 DCF: ALU_K[.FFFF],J/S11
            ;2097
U 1178. 0118.0038.1B80.0800.0000.117C   ;2098 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1179. 0118.0038.0B80.0800.0000.117C   ;2099 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 117A. 0118.0038.0780.0800.0000.117C   ;2100 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 117B. 0118.0038.C380.0800.0000.117C   ;2101 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
            ;2102
U 117C. 0100.003E.0380.0800.0000.0001   ;2103 SX: D_D.LEFT2,SI/7,RETURN1   ; EXIT
            ;2104
            ;2105
            ;2106   .TOC "   ASCII MESSAGES"
            ;2107 ;*
            ;2108 ; FOLLOWING ARE THE ASCII MESSAGES USED BY THIS MODULE
            ;2109 ;
            ;2110 ;& 1,/TEST ABORTED - NOT ENOUGH MEMORY/
            ;2111 ;& 2,/TEST ABORTED - NO DW780/
            ;2112 ;
            ;2113
            ;2114
            ;2115 ;*
            ;2116 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2117 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2118 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2119 ; TYPING IT.
            ;2120 ;-
            ;2121
U 117D. 0810.003B.4D80.0978.0000.117E   ;2122 SUBTST: D_RC[0F],KMX/.FF00
U 117E. 0019.4016.4D80.09FB.0000.0001   ;2123   RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2124
            ;2125 ;*
```

```
            ;2126 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2127 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2128 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2129 ;-
            ;2130
U 117F, 0010,0038,01C0,0978,0000,1180   ;2131 SETRCF: Q_RC[0F]
U 1180, 0019,2034,4DC0,0800,0000,1181   ;2132  Q_Q.AND.K[.FF00]
U 1181, 0019,2032,1D80,09F8,0000,0001   ;2133  RC[0F]_Q.OR.SC,RETURN1
U 1155, 0019,2000,05C0,0800,0000,12AB   ;2134 1155: Q_Q-K[.1],J/12AB
U 12AA, 0000,003C,0180,0800,0000,1182   ;2135 12AA: NOP     ; ECO LOCATION
            ;2136
            ;2137
            ;2138 DISPATIRD:
U 1182, 0000,003F,0180,0800,1000,1200   ;2139  SUB/SPEC,MSC/IRD,J/1200
            ;2140
U 1220, 0000,003E,01F8,0800,0000,0001   ;2141 1220: Q_0,RETURN1
U 1224, 0018,003A,05C0,0800,0000,0001   ;2142 1224: Q_K[.1],RETURN1
U 1228, 0018,003A,09C0,0800,0000,0001   ;2143 1228: Q_K[.2],RETURN1
U 122C, 0018,003A,0DC0,0800,0000,0001   ;2144 122C: Q_K[.3],RETURN1
U 1230, 0018,003A,11C0,0800,0000,0001   ;2145 1230: Q_K[.4],RETURN1
U 1234, 0018,003A,D5C0,0800,0000,0001   ;2146 1234: Q_K[.6],RETURN1
U 1238, 0018,003A,5DC0,0800,0000,0001   ;2147 1238: Q_K[.7],RETURN1
U 123C, 0018,003A,01C0,0800,0000,0001   ;2148 123C: Q_K[.8],RETURN1
            ;2149
            ;2150
            ;2151
            ;2152 ;*
            ;2153 ; THE FOLLOWING ROUTINES ARE USES TO SET OR CLEAR BITS IN THE CONSOLE
            ;2154 ; REGISTERS. BYTE 0 OF THE D REGISTER MUST CONTAIN "C", BYTE ONE OF THE
            ;2155 ; D REGISTER MUST CONTAIN THE CONSOLE REGISTER ID, I.E. 0=TXCS, 2=RXCS,
            ;2156 ; AND 4=MCR, AND THE UPPER WORD OF THE D REGISTER MUST CONTAIN THE DATA
            ;2157 ; TO SET IN THE REGISTER.
            ;2158 ;
            ;2159 ; THE SUBROUTINES MUST BE CALLED WITH THE DATA TO LOAD INTO THE REGISTER
            ;2160 ; IN SCRATCH PAD R[0A].
            ;2161 ;
            ;2162 ; NOTE: ONLY BIT 15, 9, AND 8 OF THE MCR REGISTER CAN BE SET OR CLEARED.
            ;2163 ;       ONLY BIT 7 OF THE RXCS REGISTER CAN BE SET OR CLEARED.
            ;2164 ;       BIT 7 OF THE TXCS REGISTER IS ALWAYS SET WHILE THE MICRO CODE IS RUNNING.
            ;2165 ;-
            ;2166
U 1183, 0818,0038,1180,0800,0000,1184   ;2167 SETMCR: D_K[.4]     ; GET REGISTER ID
U 1184, 0000,003C,01F8,0800,0084,7185   ;2168  SC_K[.8],Q_0     ; SETUP FOR SHIFT
U 1185, 0D00,003C,0180,0800,0000,118A   ;2169  D_DAL.SC,J/COMCON  ; SHIFT TO BYTE 1
            ;2170
U 1186, 0F00,003C,01F8,0800,0000,118A   ;2171 SETTXCS:D_0,J/COMCON,Q_0  ; SET REGISTER ID
U 1187, 0818,0038,0980,0800,0000,1188   ;2172 SETRXCS:D_K[.2]     ; GET REGISTER ID
U 1188, 0000,003C,01F8,0800,0084,7189   ;2173  SC_K[.8],Q_0     ; SETUP FOR SHIFT
U 1189, 0D00,003C,0180,0800,0000,118A   ;2174  D_DAL.SC,J/COMCON  ; PUT IN BYTE 1
            ;2175
U 118A, 0019,0030,8580,0AD8,0000,118B   ;2176 COMCON: R[0B]_D.OR.K[.C]  ; INSERT BYTE 0 AND SAVE
U 118B, 0800,003C,0180,0A50,0000,118C   ;2177  D_R[0A]     ; GET DATA TO PUT IN CONSOLE REG
U 118C, 0000,003C,6580,0800,0084,718D   ;2178  SC_K[.10]    ; SETUP FOR SHIFT
U 118D, 0D00,003C,0180,0A58,0000,118E   ;2179  D_DAL.SC,LAB_R[0B]  ; PUT IN UPPER WORD
U 118E, 081C,2030,0180,0800,0000,105E   ;2180  D_LA.OR.D,J/CALLCON  ; INSERT LOWER WORD AND CALL CONSOLE
```

```
;2181
;2182
```

```
            ;2183  .PAGE
            ;2184  .TOC "    RESET SUBTEST NUMBER"
            ;2185 ;++
            ;2186 ; FUNCTIONAL DESCRIPTION:
            ;2187 ;
            ;2188 ; Since some  of the tests will have to be restarted when two
            ;2189 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2190 ; the subtest counter to zero ( only for thoes tests that have
            ;2191 ; more than one subtest). This subroutine may also be necessary
            ;2192 ; when a test is being loop on.
            ;2193 ;
            ;2194 ; CALLING CONSTRAINT:
            ;2195 ;
            ;2196 ; =0
            ;2197 ; SIDE EFFECTS:
            ;2198 ;
            ;2199 ; D is destroyed
            ;2200 ;
            ;2201 ;--
            ;2202 RESET.SUBTST:
            ;2203  RC[0F]_0,    ; Reset subtest number
U 118F, 0003,003E,0180,09F8,0000,0001  ;2204   RETURN1    ; ...and return
            ;2205
            ;2206
```

```
            ;2207  .PAGE
            ;2208  .TOC "   Set Trap Mask"
            ;2209 ;++
            ;2210 ; FUNCTIONAL DESCRIPTION:
            ;2211 ;
            ;2212 ; This routine is used to set a bit in the Micro Trap Mask
            ;2213 ; R[0F].
            ;2214 ;
            ;2215 ; CALLING CONSTRAINT:
            ;2216 ;
            ;2217 ; =0
            ;2218 ;
            ;2219 ; INPUTS:
            ;2220 ;
            ;2221 ; SC - Contains bit number to set.
            ;2222 ;
            ;2223 ; OUTPUTS:
            ;2224 ;
            ;2225 ; rc[0E] - Contains the current trap mask
            ;2226 ;
            ;2227 ; SIDE EFFECTS:
            ;2228 ;
            ;2229 ; d regsiter is destroyed
            ;2230 ;--
            ;2231 SET_TRAP_MASK:
U 1190, 0800.003C.0180.0A78.0000.1191  ;2232  d_r[0f]
U 1191, 0801.001C.0180.0AF8.0000.1192  ;2233  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1192, 0001.003E.0180.0AF0.0000.0001  ;2234  r[0E]_d,return1    ; Save mask and return
            ;2235
            ;2236
```

```
            ;2237  .PAGE
            ;2238  .TOC "   Disable Cache Routine"
            ;2239 ;**
            ;2240 ; FUNCTIONAL DESCRIPTION:
            ;2241 ;
            ;2242 ; This routine disables cache hits by setting bits <16:15>
            ;2243 ; in the maintenance register.
            ;2244 ;
            ;2245 ; CALLING SEQUENCE:
            ;2246 ;
            ;2247 ; =0
            ;2248 ;
            ;2249 ; SIDE EFFECTS:
            ;2250 ;
            ;2251 ; D & Q Registers destroyed
            ;2252 ;--
            ;2253 DISABLE.CACHE:
U 1193. 0818.0038.4580.0800.0000.1194  ;2254  d_k[.8000]  ; ... and seed constant
U 1194. 0500.003C.0180.0800.0000.1195  ;2255  d_d.left  ; Generate final constant
U 1195. 0819.0030.4580.0800.0000.1196  ;2256  d_d.or.k[.8000]  ; ... = 00018000
U 1196. 0000.003E.7580.3C00.0000.0001  ;2257  id[maint]_d,return1 ; Disable cache and return
            ;2258
            ;2259 ;*
            ;2260 ; THIS SUBROUTINE EXECUTE AN SBI UNJAM SEQUENCE
            ;2261 ;-
            ;2262
            ;2263 SBI.UNJAM:
U 1197. 0F00.003C.0180.0800.0000.1198  ;2264 UNJAM: D_0
U 1198. 0000.003C.7580.3C00.0000.1199  ;2265  ID[MAINT]_D   ; RELOAD THE MAT REG
U 1199. 0F00.003C.0180.8000.0000.119A  ;2266  D_0,MCT/SBI.HOLD  ; ASSERT HOLD FOR 16 CYCLES
U 119A. 0019.0020.6180.8000.0010.119B  ;2267 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD ; DONE YET?
U 119B. 0000.013C.0180.8000.0000.1044  ;2268  Z?,MCT/SBI.HOLD   ; BRANCH IF YES
U 1044. 0819.0014.0580.8000.0000.119A  ;2269 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM0 ; CONTINUE HOLDING THE BUS
            ;2270
U 1045. 0F00.003C.0180.8800.0000.119C  ;2271  D_0,MCT/SBI.HOLD+UNJAM  ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
U 119C. 0019.0020.6180.8800.0010.119D  ;2272 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD+UNJAM ; DONE YET?
U 119D. 0000.013C.0180.8800.0000.1046  ;2273  Z?,MCT/SBI.HOLD+UNJAM  ; BRANCH IF YES
U 1046. 0819.0014.0580.8800.0000.119C  ;2274 =0 D_D+K[.1],MCT/SBI.HOLD+UNJAM,J/UNJAM1 ; CONTINUE UNJAM
            ;2275
U 1047. 0F00.003C.0180.8000.8000.119E  ;2276  D_0,MCT/SBI.HOLD,INTRPT.ACK ; ASSERT HOLD FOR 16 CYCLES
U 119E. 0019.0020.6180.8000.0010.119F  ;2277 UNJAM2: ALU_D[XOR]K[.F],MCT/SBI.HOLD,CLK.UBCC ; DONE YET?
U 119F. 0000.013C.0180.8000.0000.1048  ;2278  Z?,MCT/SBI.HOLD   ; BRANCH IF YES
U 1048. 0819.0014.0580.8000.0000.119E  ;2279 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM2 ; CONTINUE HOLDING
            ;2280
U 1049. 0000.003C.8580.0800.0084.71A0  ;2281  SC_K[.C]
U 11A0. 0803.001C.0180.0800.4000.11A1  ;2282  D_NOT.MASK,INTRPT.STROBE
U 11A1. 0000.003E.7580.3C00.0000.0001  ;2283  ID[MAINT]_D,RETURN1  ; TURN OFF THE SBI
            ;2284
            ;2285
```

```
            ;2286  .PAGE
            ;2287  .TOC "   Initialize the SBI"
            ;2288 ;**
            ;2289 ; FUNCTIONAL DESCRIPTION:
            ;2290 ;
            ;2291 ; This routine performs the following functions:
            ;2292 ;
            ;2293 ;  Executes an SBI Unjam
            ;2294 ;  Clears the SBI Fault Latch
            ;2295 ;  Clears the SBI Error Register
            ;2296 ;
            ;2297 ; CALLING CONSTRAINT:
            ;2298 ;
            ;2299 ; =0
            ;2300 ;
            ;2301 ; SIDE EFFECTS:
            ;2302 ;
            ;2303 ; D & Q Register destroyed
            ;2304 ;--
            ;2305 =0
            ;2306 SBI.INIT:
U 104A, 0000,003D,0180,0800,0000,1197  ;2307  call[sbi.unjam]  ; Unjam the SBI
U 104B, 0818,0038,C180,0800,0000,11A2  ;2308  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11A2, 0801,0028,6580,3C00,0000,11A3  ;2309  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11A3, 0F00,003C,6D80,3C00,0000,11A4  ;2310  id[fault]_d,d_0
U 11A4, 0000,003C,6D80,3C00,0000,11A5  ;2311  id[fault]_d
U 11A5, 0000,003E,6580,3C00,0000,0001  ;2312  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2313
            ;2314
            ;2315 ;*
            ;2316 ; THE FOLLOWING SUBROUTINE GENERATES CONSTANTS AND SAVES THEM IN THE RA
            ;2317 ; SCRATCH PADS. FOLLOWING ARE THE CONSTANTS GENERATED:
            ;2318 ;
            ;2319 ; R[2] = FORCE REPLACE GROUP 1 AND FORCE MISS GROUP 0
            ;2320 ; R[3] = R[6] PLUS DISABLE SBI CYCLE
            ;2321 ; R[4] = CPU ADDRESS FOR AUTO REFILL (0)
            ;2322 ; R[5] = IB ADDRESS FOR AUTO REFILL (3FC)
            ;2323 ; R[6] = FORCE REPLACE GROUP 0 AND FORCE MISS GROUP 1
            ;2324 ; R[7] = GROUP 0 AND GROUP 1 MATCH BIT MASK
            ;2325 ; R[8] = GROUP 0 MATCH BIT
            ;2326 ; R[9] = GROUP 1 MATCH BIT
            ;2327 ; R[A] = R[6] PLUS REVERSE PARITY GROUP 0, BYTE 0 (DATA)
            ;2328 ; R[B] = TIMEOUT ADDRESS ( = 01000000 )
            ;2329 ; R[C] = SILO COUNTER MASK
            ;2330 ; R[D] = ADDRESS TO FORCE RDS (4020)
            ;2331 ; R[E] = ADDRESS OF MEMORY CONFIGURATION REGISTER B
            ;2332 ;-
            ;2333
            ;2334 =0
            ;2335 MAINTREGSETUP:
U 104C, 0000,003D,F580,0800,0084,717F  ;2336  SETRCF[.A]  ; Set the Argument count to CONSOLE
U 104D, 0000,003C,0180,0800,0000,1050  ;2337  NOP
U 1050, 0000,003D,0180,0800,0000,11C6  ;2338 =0 CALL[CONFIG.MS780E]  ; Find MS780E memory and configure it
U 1051, 0810,0038,0180,0970,0000,11A6  ;2339  D_RC[0E]  ; Get CNFG Reg A Address
U 11A6, 0019,0014,1180,0AF0,0000,11A7  ;2340  R[0E]_D+K[.4]  ; Make it in CNFG B Address
```

```
U 11A7, 0818,0038,8580,0800,0000,11A8  ;2341  D_K[.C]    ; Form "FORCE RDS" constant for
        ;2342       ; ...CNFG Reg B in R[0D]
        ;2343       ; ...First ECC substitute bits for
        ;2344       ; ...data of zero
U 11A8, 0000,003C,F580,0800,0084,71A9  ;2345  SC_K[.A]   ; And now the Diagnostic Mode bits
        ;2346  D_D.ORNOT.MASK,    ; ...and save them in
U 11A9, 0801,001C,0180,0AE8,0000,1052  ;2347   R[0D]_ALU  ; ...R[0D]
U 1052, 0000,003D,0180,0800,0000,104A  ;2348 =0 CALL[SBI.INIT]   ; Initialize the SBI
U 1053, 0000,003C,0180,0800,0000,11AA  ;2349  NOP
U 11AA, 0003,003C,0180,0AA0,0000,1054  ;2350  R[4]_0
U 1054, 0818,0039,5580,0800,0000,1174  ;2351 =0 D_K[.3F],CALL,J/DCC  ; GENERATE 3FC
U 1055, 0001,003C,0180,0AA8,0000,1056  ;2352  R[5]_D     ; SAVE
U 1056, 0818,0039,7580,0800,0000,1168  ;2353 =0 D_K[.20],CALL,J/DC0  ; GENERATE 200(X)
U 1057, 0001,003C,0180,0AC0,0000,11AB  ;2354  R[8]_D     ; GROUP 0 MATCH BIT
U 11AB, 0501,003C,0180,0AB8,0000,11AC  ;2355  R[7]_D,D_D.LEFT   ; PART OF MASK
U 11AC, 0001,003C,0180,0AC8,0000,11AD  ;2356  R[9]_D     ; GROUP 1 MATCH BIT
U 11AD, 0000,003C,0180,0A38,0000,11AE  ;2357  LAB_R[7]
U 11AE, 001C,2030,0180,0AB8,0000,11AF  ;2358  R[7]_LA.OR.D   ; COMPLETE THE MASK
U 11AF, 0100,003C,0180,0800,0000,11B0  ;2359  D_D.LEFT2
U 11B0, 0500,003C,0180,0800,0000,11B1  ;2360  D_D.LEFT    ; GENERATE FR GR0 AND FM GR1
U 11B1, 0501,003C,0180,0A90,0000,11B2  ;2361  R[2]_D,D_D.LEFT
U 11B2, 0500,003C,01E0,0800,0000,11B3  ;2362  Q_D,D_D.LEFT    ; ...
U 11B3, 081D,0030,01E0,0AB0,0000,11B4  ;2363  R[6]_D.OR.Q,D_ALU,Q_D   ; SAVE
U 11B4, 0000,003C,7580,3C00,0000,11B5  ;2364  ID[MAINT]_D    ; SETUP THE MAINTENANCE REGISTER
U 11B5, 0000,003C,8580,0800,0084,71B6  ;2365  SC_K[.C]    ; SETUP TO GENERATE BIT 12
U 11B6, 0801,001C,0180,0800,0000,11B7  ;2366  D_D.ORNOT.MASK    ; INSERT BIT 12
U 11B7, 0001,003C,0180,0A98,0000,11B8  ;2367  R[3]_D     ; SAVE
U 11B8, 0C00,003C,0180,0A10,0000,11B9  ;2368  LAB_R[2],D_Q    ; COMPLETE DATA FOR R2
U 11B9, 0500,003C,0180,0800,0000,11BA  ;2369  D_D.LEFT
U 11BA, 001C,2030,0180,0A90,0000,11BB  ;2370  R[2]_LA.OR.D    ;
U 11BB, 0818,0038,5180,0800,0000,11BC  ;2371  D_K[.1E]    ; GENERATE CONSTANT FOR R[A]
U 11BC, 0000,003C,65F8,0800,0084,71BD  ;2372  SC_K[.10],Q_0    ; ...
U 11BD, 0D00,003C,01C0,0A30,0000,11BE  ;2373  D_DAL.SC,Q_R[6]    ; ...
U 11BE, 001D,0030,0180,0AD0,0000,11BF  ;2374  R[0A]_D.OR.Q    ; SAVE
U 11BF, 0818,0038,0580,0800,0000,11C0  ;2375  D_K[.1]     ; GENERATE A TIMEOUT ADDRESS
U 11C0, 0B00,003C,0180,0800,0000,11C1  ;2376  D_D.SWAP    ; ...
U 11C1, 0001,003C,0180,0AD8,0000,11C2  ;2377  R[0B]_D     ; SAVE
U 11C2, 0818,0038,61F8,0800,0000,11C3  ;2378  D_K[.F],Q_0
U 11C3, 0D00,003C,0180,0800,0000,11C4  ;2379  D_DAL.SC
U 11C4, 0001,003C,0180,0AE0,0000,11C5  ;2380  R[0C]_D     ; SILO COUNTER MASK
U 11C5, 0000,003E,0180,0890,0000,0001  ;2381  RETURN1
        ;2382
```

```
;2383  .PAGE
;2384  .TOC "   CONFIGURE MS780-E/H"
;2385 ;*************************************************************
;2386 ;**
;2387 ;
;2388 ; Functional Description:
;2389 ; ----------------------
;2390 ;
;2391 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2392 ; specifically  check  the  memory, but  make use of it to execute.
;2393 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2394 ; that  if  a  Read/Write operation will take place no false errors
;2395 ; will be logged due to misconfigured memory.
;2396 ;
;2397 ; Calling Constraint: =00
;2398 ; ------------------
;2399 ;
;2400 ;
;2401 ; Inputs:
;2402 ; ------
;2403 ;
;2404 ; RC[0E] - I/O BASE OF MS780-E/H
;2405 ;
;2406 ;
;2407 ; Outputs:
;2408 ; -------
;2409 ;
;2410 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2411 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2412 ;      OR IF NO MS780-E/Hs WERE FOUND
;2413 ;      ON THE SBI
;2414 ;
;2415 ; Side Effects:
;2416 ; ------------
;2417 ;
;2418 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2419 ;
;2420 ;
;2421 ;
;2422 ;--
;2423 ;*************************************************************
;2424 CONFIG.MS780E:
U 11C6, 0818,0038,0D80,0800,0000,11C7  ;2425  D_K[.3]    ; Set up flag for the Fail Chain
U 11C7, 0001,003C,0180,09E8,0000,1010  ;2426  RC[0D]_D    ; ...
U 1010, 0000,003D,0180,0800,0000,1058  ;2427 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1011, 0000,003D,0980,0800,0084,7040  ;2428  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 1012, 0000,003C,0180,0800,0000,1020  ;2429  NOP     ; OK. Found at least one MS780-E/H
;2430 =
;2431 CONFIG.RETRY:     ; Entry point for the case in which the
;2432       ; ...first MS780-E/H could not be
;2433       ; ...properly configured
U 1020, 0000,003D,0180,0800,0000,1038  ;2434 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1021, 0000,003C,0180,0800,0000,1028  ;2435  J/CNFG.LMIS   ; Lower controller misconfigured
U 1022, 0000,003E,0180,0800,0000,0001  ;2436  RETURN1     ; OK. Lower Controller is configured
;2437 =
```

```
            ;2438 =00
            ;2439 CNFG.LMIS:
U 1028, 0000,003D,0180,0800,0000,1034  ;2440  CALL[SELECT.UPPER]  ; Select the upper controller
U 1029, 0000,003C,0180,0800,0000,1030  ;2441  J/CNFG.UMIS    ; Upper Controller Misconfigured
U 102A, 0000,003E,0180,0800,0000,0001  ;2442  RETURN1     ; OK. Upper Controller is configured
            ;2443 =
            ;2444 =00
            ;2445 CNFG.UMIS:
U 1030, 0000,003D,0180,0800,0000,11FB  ;2446  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2447       ; ...MS780-E/H found so go and see
            ;2448       ; ...if there are any more
U 1031, 0000,003C,0180,0800,0000,11C8  ;2449  J/CONFIG.RETRY    ; Found another MS780-E/H. Go and
            ;2450       ; ...try to configure it
U 1032, 0818,0038,0D80,0800,0000,1033  ;2451  D_K[.3]     ; Set Flag for Fail Chain
U 1033, 0001,003C,0180,09E8,0000,11C8  ;2452  RC[0D]_D     ; ...
U 11C8, 0000,003D,0980,0800,0084,7040  ;2453  ERROR1[.2]    ; Could not configure any MS780-E/H
            ;2454       ; ...abort the test
            ;2455 =
            ;2456
            ;2457
```

```
            ;2458  .PAGE
            ;2459  .TOC "   Find MS780-E Memory"
            ;2460 ;++
            ;2461 ; FUNCTIONAL DESCRIPTION:
            ;2462 ;
            ;2463 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2464 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2465 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2466 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2467 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2468 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2469 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2470 ;     (zero).
            ;2471 ;     All the other MS780-E/H's found will have their starting  address
            ;2472 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2473 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2474 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2475 ;     current MS780-E unit under test.
            ;2476 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2477 ;     set to maximum.
            ;2478 ;
            ;2479 ; CALLING CONSTRAINT:
            ;2480 ;
            ;2481 ; =00
            ;2482 ;
            ;2483 ; OUTPUTS:
            ;2484 ;
            ;2485 ; rc[0e] - Contains address of Configuration Register
            ;2486 ; r[0]  - Contains TR level
            ;2487 ;
            ;2488 ; SIDE EFFECTS:
            ;2489 ;
            ;2490 ; D register is destroyed.
            ;2491 ;--
            ;2492 =0
            ;2493 FIND.MS780E:
U 1058, 0000.003D.0180.0800.0000.104A  ;2494  call[sbi.init]  ; Make sure SBI is enabled
U 1059, 0000.003C.0180.0800.0000.105C  ;2495  nop
U 105C, 0000.003D.0180.0800.0000.1193  ;2496 =0 call[disable.cache] ; Disable the cache and re-enable SBI
U 105D, 0003.003C.0180.0988.0000.11C9  ;2497  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 11C9, 0818.0038.1980.0800.0000.11CA  ;2498  d_k[zero]  ; Clear MS780-E/H counter
U 11CA, 0000.003C.F980.3C00.0000.11CB  ;2499  id[3e]_d  ; ...
U 11CB, 0018.0038.0580.0A80.0000.11CC  ;2500  r[0]_k[.1]  ; Init TR counter
U 11CC, 0F03.003C.0180.09F0.0000.1060  ;2501  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2502      ; ...CNFG A Reg
            ;2503 =0
            ;2504 FIND.MEM.LOOP:
U 1060, 0800.003D.0180.0A00.0000.11F2  ;2505  d_ri[0],call[generate.io]; Generate address of TR level
U 1061, 0001.003C.0180.09F0.0200.1062  ;2506  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1062, 0000.003D.8980.0800.0084.7190  ;2507 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2508  d[long]_cache.p, ; Read the specified tr level
U 1063, 0000.003C.0180.C970.0000.11CD  ;2509   lc_rc[0e]  ; ...
U 11CD, 0000.003C.0180.0A78.0010.11CE  ;2510  alu_r[0f],clk.ubcc ; Check for no response
U 11CE, 0001.013C.0180.0880.0082.1064  ;2511  z?,sc_d,la_ra[0] ; Branch if no adapter here
U 1064, 0000.003C.0180.0800.0000.11CF  ;2512 =0 j/check.adpt.code ; Check for a memory
```

```
U 1065, 0000,003C,0180,0800,0000,11EE  ;2513  j/find.mem.lp1  ; Try next tr
        ;2514 CHECK.ADPT.CODE:
U 11CF, 0000,003C,1D80,0800,1404,71D0  ;2515  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 11D0, 0000,163C,0180,0800,0000,1008  ;2516  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,71D1  ;2517 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,71D2  ;2518  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,11EE  ;2519  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,11EE  ;2520  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,71D7  ;2521  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,11EE  ;2522  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,11DB  ;2523  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,11DB  ;2524  rc[0e]_lc,j/found.ms780e ; MS780-E
        ;2525 ;
        ;2526 ; Found an MS780-A or -C. Set the starting address
        ;2527 ; to maximum.
        ;2528 ;
U 11D1, 0818,0038,5D80,0800,0000,11D3  ;2529 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 11D2, 0818,0038,0580,0800,0000,11D3  ;2530 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
        ;2531 MS780.COMMON:
U 11D3, 0819,0030,7180,0800,0000,11D4  ;2532  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 11D4, 0000,003C,01F8,0803,0080,D1D5  ;2533  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 11D5, 0D00,003C,0180,0800,0000,11D6  ;2534  d_dal.sc  ; Put data in bits<29:14>
        ;2535  cache.p_d[long], ; Set the starting address to maximum
U 11D6, 0000,003C,0180,A800,0000,11EE  ;2536       j/find.mem.lp1 ; and continue TR scan
        ;2537 ;
        ;2538 ; Found an MA780. Set the starting address to maximum
        ;2539 ;
        ;2540 MA780.MAX:
U 11D7, 0818,0038,39F8,0803,0000,11D8  ;2541  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 11D8, 0D00,003C,0180,0803,0000,11D9  ;2542  d_dal.sc,va_va+4 ; Put in proper position
U 11D9, 0000,003C,0180,0803,0000,11DA  ;2543  va_va+4   ; Point to Port Invalidate Ctrl Register
        ;2544  cache.p_d[long], ; Set starting address to maximum
U 11DA, 0000,003C,0180,A800,0000,11EE  ;2545       j/find.mem.lp1
        ;2546 ;
        ;2547 ; Found an MS780E
        ;2548 ;
        ;2549 FOUND.MS780E:
U 11DB, 0000,003C,F9F0,2C00,0000,11DC  ;2550  q_id[3e]  ; Set up to see how many MS780-E's
        ;2551  alu_q.xor.k[.3], ; Are there Maximum units?
U 11DC, 0019,2020,0D80,0800,0010,11DD  ;2552   clk.ubcc ; Check condition codes
U 11DD, 0000,013C,0180,0800,0000,1066  ;2553  z?    ; Are we at maximum units for system
U 1066, 0000,003C,0180,0800,0000,11DE  ;2554 =0 J/ms780e.tst  ; No go find how many units ther are
U 1067, 0818,0038,C180,0800,0000,1068  ;2555  d_k[.ffff]  ; Yes set address to maximum
U 1068, 0000,003D,0180,0800,0000,11F6  ;2556 =0 call[set.start.addr] ; .....
U 1069, 0000,003C,0180,0800,0000,11EE  ;2557  j/find.mem.lp1  ; Go check to see if we are done
        ;2558 ;
        ;2559 MS780E.TST:
        ;2560  alu_rc[01],  ; Check "Found MS780E Flag"
U 11DE, 0010,0038,0180,0908,0010,11DF  ;2561   clk.ubcc ; Check condition codes
U 11DF, 0810,0138,0180,0970,0000,106A  ;2562  z?,d_rc[0e]  ; Is this unit the first MS780E ?
        ;2563 =0 j/not.first.ms780e, ; No
U 106A, 0818,0038,C180,0800,0000,1072  ;2564   d_k[.ffff] ; Set starting address
U 106B, 0001,003C,0180,0988,0000,11E0  ;2565  rc[01]_d  ; Yes put base address in rc[01]
U 11E0, 0818,0038,7980,0800,0000,11E1  ;2566  d_k[.30]  ; Set up SC to point to first unit
U 11E1, 0019,0014,F580,0800,0082,11E2  ;2567  alu_d+k[.a],sc_alu ; ...
```

```
U 11E2. 0810.0038.0180.0908.0000.11E3  ;2568  d_rc[01]  ; Put base address of unit in D
U 11E3. 0000.003C.0180.3400.0000.11E4  ;2569  id(sc)_d  ; Put base address in ID pointed to by SC
U 11E4. 0F00.003C.0180.0800.0000.1070  ;2570  d_0   ; Prepare to set starting address to Zero
U 1070. 0000.003D.0180.0800.0000.11F6  ;2571 =0 call[set.start.addr] ; Go do it
          ;2572  j/find.mem.lp1,  ; Check to see if we are done
U 1071. 0003.003C.0180.09E8.0000.11EE  ;2573   rc[0d]_0 ; ....
          ;2574 =0
          ;2575 NOT.FIRST.MS780E:
U 1072. 0000.003D.0180.0800.0000.11F6  ;2576  call[set.start.addr] ; Go set starting address to maximum
U 1073. 0000.003C.F9F0.2C00.0000.11E5  ;2577  q_id[3e]  ; Get the number of MS780-Es found
U 11E5. 0819.2014.0580.0800.0000.11E6  ;2578  d_q+k[.1]  ; Increment this counter
U 11E6. 0000.003C.F980.3C00.0000.11E7  ;2579  id[3e]_d  ; Save the counter
U 11E7. 0018.0038.11C0.0800.0000.11E8  ;2580  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2581     ; ...were the I/O base addresses will be stored
U 11E8. 081D.2000.7980.0800.0000.11E9  ;2582  d_q-d,k[.30]  ; ... adjust pointer
U 11E9. 0819.0014.7980.0800.0000.11EA  ;2583  d_d+k[.30]  ; Point the SC to the ID register
U 11EA. 0000.003C.F580.0800.0000.11EB  ;2584  k[.a]   ; ...to receive I/O base address
U 11EB. 0019.0014.F580.0800.0082.11EC  ;2585  alu_d+k[.a],sc_alu ; ...
U 11EC. 0810.0038.0180.0970.0000.11ED  ;2586  d_rc[0e]  ; Get base address to d
U 11ED. 0000.003C.0180.3400.0000.11EE  ;2587  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2588
          ;2589 ;
          ;2590 ; Try next tr
          ;2591 ;
          ;2592 FIND.MEM.LP1:
U 11EE. 0818.0014.0580.0A80.0000.11EF  ;2593  r[0]_le+k[.1],d_alu ; Increment TR level
          ;2594  alu_d.and.k[.f],
U 11EF. 0019.0034.6180.0800.0010.11F0  ;2595       clk.ubcc  ; Check if all done
U 11F0. 0000.013C.0180.0800.0000.1074  ;2596  z?   ; Branch if yes
U 1074. 0000.003C.0180.0800.0000.1060  ;2597 =0 j/find.mem.loop  ; Try next TR
U 1075. 0810.0038.0180.0908.0010.11F1  ;2598  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 11F1. 0000.013C.0180.0800.0000.1076  ;2599  z?   ; Check condition codes
U 1076. 0001.003E.0180.09F0.0000.0002  ;2600 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1077. 0000.003E.0180.0800.0000.0001  ;2601  return1   ; No MS780E found
          ;2602
```

```
            ;2603  .PAGE
            ;2604  .TOC "   Generate IO Address"
            ;2605 ;**
            ;2606 ; FUNCTIONAL DESCRIPTION:
            ;2607 ;
            ;2608 ; This routine is used to generate an SBI Configuration Register
            ;2609 ; address from a TR level.
            ;2610 ;
            ;2611 ; CALLING CONSTRAINT:
            ;2612 ;
            ;2613 ; =0
            ;2614 ;
            ;2615 ; INPUTS:
            ;2616 ;
            ;2617 ; D - Contains TR level
            ;2618 ;
            ;2619 ; OUTPUTS:
            ;2620 ;
            ;2621 ; D - Contains SBI IO address
            ;2622 ;--
            ;2623 GENERATE.IO:
U 11F2, 0000,003C.89F8,0800.0084,71F3  ;2624  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 11F3, 0D00,003C.8D80,0800.0084,71F4  ;2625  d_del.sc,sc_k[.1f] ; Put TR level in place, setup to
U 11F4, 0000,003C.0980,0800.0084,B1F5  ;2626  sc_sc-k[.2]  ; insert bit 29
U 11F5, 0801,001E.0180,0800.0000,0001  ;2627  d_d.ornot.mask,return1 ; and return
            ;2628
            ;2629
```

```
            ;2630  .PAGE
            ;2631  .TOC "   Set Starting Address"
            ;2632 ;**
            ;2633 ; FUNCTIONAL DESCRIPTION:
            ;2634 ;
            ;2635 ; This routine is used to set the starting address of the
            ;2636 ; memory to the specified value.
            ;2637 ;
            ;2638 ; CALLING CONSTRAINT:
            ;2639 ;
            ;2640 ; =0
            ;2641 ;
            ;2642 ; INPUTS:
            ;2643 ;
            ;2644 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2645 ;     (B Register Image divided by 2**16)
            ;2646 ; rc[0] - Contains Address of Register A
            ;2647 ;
            ;2648 ; OUTPUTS:
            ;2649 ;
            ;2650 ; d - Contains aligned starting address (B Register Image)
            ;2651 ;
            ;2652 ; SIDE EFFECTS:
            ;2653 ;
            ;2654 ; d,q,va, and sc are destroyed
            ;2655 ;--
            ;2656 SET.START.ADDR:
U 11F6, 0010,0038,6580,0970,0284,71F7  ;2657  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11F7, 0000,003C,01F8,0803,0000,11F8  ;2658  va_va+4,q_0  ; Set address to Register B
            ;2659  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11F8, 0D00,003C,0980,0800,0084,B1F9  ;2660       sc_sc-k[.2]  ; Setup to insert bit 14
U 11F9, 0801,001C,0180,0800,0000,11FA  ;2661  d_d.ornot.mask  ; Insert Write Enable bit
U 11FA, 0000,003E,0180,A800,0000,0001  ;2662  cache.p_d[long],return1 ; Set the starting address and return
            ;2663
```

```
            ;2664  .PAGE
            ;2665  .TOC "   ENABLE NEXT MS780-E"
            ;2666 ;**
            ;2667 ; FUNCTIONAL DESCRIPTION:
            ;2668 ;
            ;2669 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2670 ;    They will be tested  separately, according to the TR level at which
            ;2671 ;    they are located,  starting with the one  at the lowest level first.
            ;2672 ;          When  a test has finished  with the first unit, it will have to call
            ;2673 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2674 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2675 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2676 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2677 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2678 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2679 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2680 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2681 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2682 ;
            ;2683 ; CALLING CONSTRAINT:
            ;2684 ;
            ;2685 ;  =00
            ;2686 ;
            ;2687 ;--
            ;2688 ENABLE.NEXT.MS780E:
U 11FB. 0000.003C.F9F0.2C00.0000.11FC  ;2689  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2690  ALU_Q,   ; Check to see if it is zero
U 11FC. 0001.203C.0180.0800.0010.11FD  ;2691     CLK.UBCC  ; Check Condition Codes
U 11FD. 0000.013C.0180.0800.0000.1080  ;2692  Z?   ; ...for zero
U 1080. 0000.003C.0180.0800.0000.11FE  ;2693 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1081. 0000.003E.0180.0800.0000.0002  ;2694  RETURN2   ; Zero No more units to test
            ;2695 ENABLE.NEXT:
U 11FE. 0818.0038.C180.0800.0000.1082  ;2696  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1082. 0000.003D.0180.0800.0000.11F6  ;2697 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1083. 0818.0038.7980.0800.0000.11FF  ;2698  D_K[.30]  ; Prepare to point to the ID register holding
            ;2699     ; ...the I/O Base address of the next MS780-E
U 11FF. 0819.0014.1180.0800.0000.1200  ;2700  D_D+K[.4]  ; ...
U 1200. 0000.003C.F580.0800.0000.1201  ;2701  K[.A]   ; ...
U 1201. 0819.0014.F580.0800.0000.1202  ;2702  D_D+K[.A]  ; ...
U 1202. 0000.003C.F9F0.2C00.0000.1203  ;2703  Q_ID[3E]  ; Get MS780-E Counter
            ;2704  D_D-Q,   ; D now holds the pointer to the ID register
U 1203. 081D.0000.0180.0800.0082.1204  ;2705   SC_ALU  ; ...
U 1204. 0000.003C.01F0.2400.0000.1205  ;2706  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1205. 0001.203C.0180.09F0.0000.1206  ;2707  RC[0E]_Q  ; Save it also in RC[0E]
U 1206. 0F03.003C.0180.09E8.0000.1084  ;2708  RC[0D]_0,D_0  ; Set starting address to zero
U 1084. 0000.003D.0180.0800.0000.11F6  ;2709 =0 CALL[SET.START.ADDR] ; ....
U 1085. 0F00.003C.F9F0.2C00.0000.1207  ;2710  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1207. 081D.2008.0180.0800.0000.1208  ;2711  D_Q-D-1   ; Subtract 1 from total
U 1208. 0000.003C.F980.3C00.0000.1086  ;2712  ID[3E]_D  ; Adjust the pointer
U 1086. 0000.003D.0180.0800.0000.118F  ;2713 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 1087. 0000.003E.0180.0800.0000.0001  ;2714  RETURN1   ; Tell caller another unit was found
            ;2715
```

```
            ;2716  .PAGE
            ;2717  .TOC "    Select Controller"
            ;2718 ;++
            ;2719 ; FUNCTIONAL DESCRIPTION:
            ;2720 ;
            ;2721 ; This routine is used to put the memory system into the
            ;2722 ; specified configuration with respect to controller select.
            ;2723 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2724 ; a RETURN2 is made.
            ;2725 ;
            ;2726 ; CALLING CONSTRAINT:
            ;2727 ;
            ;2728 ; =00
            ;2729 ;
            ;2730 ; INPUTS:
            ;2731 ;
            ;2732 ; d - Contains value to write to bits<2:0> of Register A
            ;2733 ; rc[0e] - Address of Register A
            ;2734 ;
            ;2735 ; SIDE EFFECTS:
            ;2736 ;
            ;2737 ; sc,d,va are destroyed
            ;2738 ;--
            ;2739 SELECT.CNTRLR:
U 1209, 0010,0038,0180,0970,0284,720A  ;2740  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 120A, 0801,001C,0180,0800,0000,120B  ;2741  d_d.ornot.mask  ; Insert the enable bit into D reg
U 120B, 0000,003C,0180,A800,0000,120C  ;2742  cache.p_d[long]  ; Write the configuration Register
U 120C, 0818,0038,5D80,0800,0000,120D  ;2743  d_k[.7]   ; Get Misconfiguration bits
U 120D, 0000,003C,61F8,0800,0084,720E  ;2744  sc_k[.F],q_0  ; ...
U 120E, 0D00,003C,0180,0800,0000,120F  ;2745  d_dal.sc  ; ... D = x38000
U 120F, 0000,003C,01E0,0800,0000,1210  ;2746  q_d   ; Save mask
U 1210, 0000,003C,0180,C800,0000,1211  ;2747  d[long]_cache.p  ; Read CNFG A Reg and
            ;2748  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1211, 001D,2034,0180,0800,0010,1212  ;2749   clk.ubcc ; ...
U 1212, 0000,013C,0180,0800,0000,1088  ;2750  z?   ; Branch if no
U 1088, 0000,003E,0180,0800,0000,0001  ;2751 =0 RETURN1   ; Bad configuration
U 1089, 0000,003E,0180,0800,0000,0002  ;2752  RETURN2   ; Configuration ok
            ;2753
            ;2754
```

```
            ;2755  .PAGE
            ;2756  .TOC "    SELECT UPPER CONTROLLER"
            ;2757 ;*************************************************************
            ;2758 ;**
            ;2759 ;
            ;2760 ; Functional Description:
            ;2761 ; ----------------------
            ;2762 ;
            ;2763 ; This subroutine can be called to select the upper
            ;2764 ; Controller on a MS780-E/H.
            ;2765 ; If the Upper controller is misconfigured then a
            ;2766 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2767 ;
            ;2768 ; Calling Constraint: =00
            ;2769 ; ------------------
            ;2770 ;
            ;2771 ;
            ;2772 ; Inputs:
            ;2773 ; ------
            ;2774 ;
            ;2775 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2776 ;
            ;2777 ; Outputs:
            ;2778 ; -------
            ;2779 ;
            ;2780 ; RC[0D] will have the address of CNFG Register D
            ;2781 ;        ( 2000x010 )
            ;2782 ;
            ;2783 ; Side Effects:
            ;2784 ; ------------
            ;2785 ;
            ;2786 ; Contents of the following registers will lost:
            ;2787 ;
            ;2788 ; D
            ;2789 ;
            ;2790 ; The Upper controller on the MS780-E/H will be configured
            ;2791 ; when the subroutine is exitted with a RETURN2
            ;2792 ;--
            ;2793 ;*************************************************************
            ;2794 =00
            ;2795 SELECT.UPPER:
            ;2796  D_K[.2],   ; Interleave mode value for CNFG Reg A
U 1034, 0818,0039,0980,0800,0000,1209  ;2797   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1035, 0000,003E,0180,0800,0000,0001  ;2798  RETURN1    ; Upper Controller Misconfigured
U 1036, 0810,0038,0180,0970,0000,1037  ;2799  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;2800  RC[0D]_D+K[.C],   ; Set the address of CNFG Reg D
U 1037, 0019,0016,8580,09E8,0000,0002  ;2801   RETURN2   ; Let caller know that the controller
            ;2802      ; ...was configured properly
            ;2803
```

```
            ;2804  .PAGE
            ;2805  .TOC "    SELECT LOWER CONTROLLER"
            ;2806 ;*************************************************************
            ;2807 ;++
            ;2808 ;
            ;2809 ; Functional Description:
            ;2810 ; ----------------------
            ;2811 ;
            ;2812 ; This subroutine can be called to select the lower
            ;2813 ; Controller on a MS780-E/H.
            ;2814 ; If the Lower controller is misconfigured then a
            ;2815 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2816 ;
            ;2817 ; Calling Constraint: =00
            ;2818 ; ------------------
            ;2819 ;
            ;2820 ;
            ;2821 ; Inputs:
            ;2822 ; ------
            ;2823 ;
            ;2824 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2825 ;
            ;2826 ; Outputs:
            ;2827 ; -------
            ;2828 ;
            ;2829 ; RC[0D] will have the address of CNFG Register C
            ;2830 ;        ( 2000x008 )
            ;2831 ;
            ;2832 ; Side Effects:
            ;2833 ; ------------
            ;2834 ;
            ;2835 ; Contents of the following registers will lost:
            ;2836 ;
            ;2837 ; D
            ;2838 ;
            ;2839 ; The Lower controller on the MS780-E/H will be configured
            ;2840 ; when the subroutine is exitted with a RETURN2
            ;2841 ;--
            ;2842 ;*************************************************************
            ;2843 =00
            ;2844 SELECT.LOWER:
            ;2845  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1038, 0818,0039,1980,0800,0000,1209  ;2846   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1039, 0000,003E,0180,0800,0000,0001  ;2847  RETURN1    ; Lower Controller Misconfigured
U 103A, 0810,0038,0180,0970,0000,103B  ;2848  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2849  RC[0D] D+K[.8],    ; Set the address of CNFG Reg D
U 103B, 0019,0016,0180,09E8,0000,0002  ;2850   RETURN2    ; Let caller know that the controller
            ;2851       ; ...was configured properly
            ;2852
            ;2853 ;+
            ;2854 ; THIS ROUTINE LOOKS FOR A DEVICE ON THE SBI AND RETURNS WITH THE DEVICES
            ;2855 ; BASE ADDRESS IN R[0] IF IT FINDS ONE, OTHERWISE R[0] IS CLEAR.
            ;2856 ;-
            ;2857
            ;2858 LOOKFORDEVICE:
```

```
U 1213. 0818.0038.7580.0800.0000.1214  ;2859  D_K[.20]
U 1214. 0B00.003C.0180.0800.0000.1215  ;2860  D_D.SWAP   ; GENERATE HIGH WORD OF DEVICE ADDRESS
U 1215. 0001.003C.0180.0A80.0000.1216  ;2861  R[0]_D    ; SAVE
U 1216. 0818.0038.75F8.0800.0000.1217  ;2862  D_K[.20],Q_0   ; GENERATE INITIAL LOW WORD
U 1217. 0000.003C.0180.0800.0084.7218  ;2863  SC_K[.8]   ; ...
U 1218. 0D00.003C.0180.0800.0000.1219  ;2864  D_DAL.SC   ; ...
U 1219. 081C.2030.0180.0A00.0000.121A  ;2865  D_D.OR.R[0]   ; D= 20002000
U 121A. 0001.003C.0180.0A80.0000.121B  ;2866  R[0]_D    ; SAVE (ADRS OF DEVICE AT TR=1)
U 121B. 0818.0038.7580.0800.0000.121C  ;2867  D_K[.20]   ; GENERATE INCREMENT FOR DEVICE ADDRESS
U 121C. 0D00.003C.0180.0800.0000.121D  ;2868  D_DAL.SC   ; D= 2000
U 121D. 0000.003C.01E0.0800.0000.121E  ;2869  Q_D    ; SAVE INCREMENT IN Q
U 121E. 0000.003C.0180.0A00.0200.121F  ;2870  VA_ALU,ALU_R[0]   ; LOAD THE DEVICE ADDRESS
U 121F. 0018.0038.6180.0A90.0000.1221  ;2871  R[2]_K[.F]   ; SET THE LOOP COUNT
U 1221. 0000.003C.8980.0800.0084.7222  ;2872  SC_K[.D]
        ;2873 TRYAGAIN:
U 1222. 0000.003C.0180.0A00.0200.1223  ;2874  VA_ALU,ALU_R[0]   ; LOAD THE ADDRESS
U 1223. 0003.001C.0180.0AF8.0000.1225  ;2875  R[0F]_NOT.MASK   ; SET THE TIMEOUT TRAP EXPECTED FLAG
U 1225. 0000.003C.0180.C800.0000.1226  ;2876  D[LONG]_CACHE.P   ; READ THE ADDRESS
U 1226. 0000.003C.0180.0A78.0010.1227  ;2877  ALU_R[0F],CLK.UBCC  ; WAS THERE A TIMEOUT?
U 1227. 0000.013C.0180.0800.0000.108A  ;2878  Z?    ; BRANCH IF YES
U 108A. 0000.003C.0180.0800.0000.1239  ;2879 =0 J/FOUNDONE   ; EXIT
        ;2880 FOUNDMEMORY:
U 108B. 0818.0038.4580.0800.0000.1229  ;2881  D_K[.8000]   ; CLEAR THE TIMEOUT BIT
U 1229. 0200.003C.0180.0800.0000.122A  ;2882  D_D.RIGHT2
U 122A. 0600.003C.0180.0800.0000.122B  ;2883  D_D.RIGHT
U 122B. 0000.003C.6580.3C00.0000.122D  ;2884  ID[SBI.ERR]_D   ; ...
U 122D. 0800.003C.0180.0A00.0000.122E  ;2885  D_R[0]    ; GET THE CURRENT ADDRESS
U 122E. 001D.0014.0180.0A80.0200.122F  ;2886  ALU_D+Q,VA_ALU,R[0]_ALU  ; INCREMENT THE ADDRESS
U 122F. 0800.003C.0180.0A10.0000.1231  ;2887  D_R[2]    ; GET THE LOOP COUNT
U 1231. 0019.0000.0580.0A90.0010.1232  ;2888  R[2]_D-K[.1],CLK.UBCC  ; DECREMENT AND CHECK
U 1232. 0000.013C.0180.0800.0000.108C  ;2889  Z?    ; BRANCH IF DONE
U 108C. 0000.003C.0180.0800.0000.1222  ;2890 =0 J/TRYAGAIN   ; CONTINUE LOOKING FOR DEVICE
U 108D. 0003.003C.0180.0A80.0084.7233  ;2891  R[0]_0,SC_K[.8]   ; NO DEVICE AVAILABLE
        ;2892 CLEARERROR:
U 1233. 0818.0038.65F8.0800.0000.1235  ;2893  D_K[.10],Q_0   ; SETUP TO CLEAR SBI ERROR REG
U 1235. 0D00.003C.0180.0800.0000.1236  ;2894  D_DAL.SC
U 1236. 0819.0030.3180.0800.0000.1237  ;2895  D_D.OR.K[.40]
U 1237. 0000.003E.6580.3C00.0000.0001  ;2896  ID[SBI.ERR]_D,RETURN1  ; CLEAR ERROR REG AND RETURN
        ;2897 FOUNDONE:
U 1239. 0200.003C.0180.0800.0000.123A  ;2898  D_D.RIGHT2   ; CHECK BIT 5 TO SEE IF ITS MEMORY
U 123A. 0200.003C.0180.0800.0000.123B  ;2899  D_D.RIGHT2   ; ...
U 123B. 0819.0034.5D80.0800.0000.123D  ;2900  D_D.AND.K[.7]   ; Mask out the lower Three bits
U 123D. 0000.193C.0180.0800.0000.1018  ;2901  D3-0?
U 1018. 0000.003C.0180.0800.0000.108B  ;2902 =1000 J/FOUNDMEMORY   ; MS780 OR MA780
U 1019. 0000.003C.0180.0800.0000.108B  ;2903  J/FOUNDMEMORY   ; MS780 OR MA780
U 101A. 0000.003C.0180.0800.0000.123E  ;2904  J/UBAMBA   ; Found DW780
U 101B. 0000.003C.0180.0800.0000.108B  ;2905  J/FOUNDMEMORY   ; Found RH780
U 101C. 0000.003C.0180.0800.0000.108B  ;2906  J/FOUNDMEMORY   ;
U 101D. 0000.003C.0180.0800.0000.108B  ;2907  J/FOUNDMEMORY   ;
U 101E. 0000.003C.0180.0800.0000.108B  ;2908  J/FOUNDMEMORY
U 101F. 0000.003C.0180.0800.0000.108B  ;2909  J/FOUNDMEMORY
        ;2910 UBAMBA:
U 123E. 0000.003C.0180.0800.0084.7233  ;2911  SC_K[.8],J/CLEARERROR
        ;2912
        ;2913
```

```
            ;2914 ;*
            ;2915 ; THIS ROUTINE IS USED TO WAIT FOR THE UNIBUS TO POWER BACK UP AFTER AN
            ;2916 ; UNJAM SEQUENCE
            ;2917 ;-
            ;2918
            ;2919 WAITFORUNIBUS:
U 123F. 0818.0038.C180.0800.0000.1240   ;2920  D_K[.FFFF]
U 1240. 0500.003C.0180.0800.0000.1241   ;2921  D_D.LEFT
U 1241. 0819.0000.0580.0800.0010.1242   ;2922 LOOPUN: D_D-K[.1],CLK.UBCC
U 1242. 0000.013C.0180.0800.0000.108E   ;2923  Z?
U 108E. 0000.003C.0180.0800.0000.1241   ;2924 =0 J/LOOPUN
U 108F. 0000.003E.0180.0800.0000.0001   ;2925  RETURN1
            ;2926
            ;2927
            ;2928 ;*
            ;2929 ; THE FOLLOWING ROUTINE CALLS THE CONSOLE TO TYPE A MESSAGE ON THE
            ;2930 ; TERMINAL. IT IS ENTERED WITH THE D REGISTER CONTAINING THE MESSAGE
            ;2931 ; NUMBER.
            ;2932 ;-
            ;2933
U 1243. 0000.003C.F580.0800.0084.7244   ;2934 MSGCOM: SC_K[.A]
            ;2935 MSGCOM_1:
U 1244. 0000.003C.0180.0800.0100.1245   ;2936  FE_SC
U 1245. 0000.003C.D9F8.0800.0084.7246   ;2937  SC_K[.9],Q_0
U 1246. 0D00.003C.0180.0800.0000.1247   ;2938  D_DAL.SC    ; PUT MSG NUMBER IN BYTE 1
U 1247. 0000.003C.C1F0.2C00.0000.1248   ;2939  Q_ID[TO]    ; TYPEOUT FLAG SET?
U 1248. 0C01.203C.01E0.0800.0010.1249   ;2940  ALU_Q,CLK.UBCC,Q_D,D_Q
U 1249. 0819.1B14.0580.0800.0000.1027   ;2941  ALU.N?,D_D+K[.1]
U 1027. 0C00.003E.0180.0800.0000.0001   ;2942 =0111 RETURN1,D_Q    ; NO TYPEOUT
U 102F. 0C00.003C.C180.3C00.0081.124A   ;2943  ID[TO]_D,D_Q,SC_FE
U 124A. 0819.0030.1D80.0800.0000.105E   ;2944  D_D.OR.K[SC],J/CALLCON  ; GO TYPE THE MESSAGE
            ;2945 MSGCOM_CRLF:
U 124B. 0018.0038.65C0.0800.0000.124C   ;2946  Q_K[.10]
U 124C. 0019.2030.0980.0800.0082.1244   ;2947  SC_Q.OR.K[.2],J/MSGCOM_1
U 124D. 0000.003C.6580.0800.0084.7244   ;2948 RCECOM: SC_K[.10],J/MSGCOM_1
            ;2949 RCECOM_CRLF:
U 124E. 0000.003C.2180.0800.0084.7244   ;2950  SC_K[.14],J/MSGCOM_1
            ;2951
            ;2952 ;**
            ;2953 ; THE FOLLOWING ROUTINE DECREMENTS THE TYPEOUT FLAG
            ;2954 ;--
U 124F. 0000.003C.C1F0.2C00.0000.1250   ;2955 DECFLG: Q_ID[TO]    ; READ THE FLAG
U 1250. 0819.2000.0580.0800.0000.1251   ;2956  D_Q-K[.1]    ; DECREMENT IT
U 1251. 0000.003E.C180.3C00.0000.0001   ;2957  ID[TO]_D,RETURN1   ; RESTORE IT AND RETURN
            ;2958
```

```
                ;2959 .PAGE "TEST 205 SBI INVALIDATE WITH GROUP 0 ADDRESS PARITY ERROR"
                ;2960 ;*************************************************************
                ;2961 ;*
                ;2962 ;
                ;2963 ; SBI INVALIDATE WITH GROUP 0 ADDRESS PARITY ERROR
                ;2964 ;
                ;2965 ; TEST DESCRIPTION
                ;2966 ;
                ;2967 ; THIS TEST USES THE MAINTENANCE SBI INVALIDATE FUNCTION TO CHECK
                ;2968 ; THAT THE CACHE WRITE PULSE IS GENERATED IF THE LOCATION TO BE
                ;2969 ; INVALIDATED CONTAINS A PARITY ERROR.
                ;2970 ;
                ;2971 ; THIS IS DONE WITH THREE CACHE ADDRESSES (1000, 40000, AND 0)
                ;2972 ; TO CHECK A PARITY ERROR IN EACH BYTE. THE ADDRESS IS FIRST WRITTEN
                ;2973 ; VALID, THE REVERSE PARITY BITS AND SBI INVALIDATE BITS
                ;2974 ; ARE SET, AND LOCATION 0 IS WRITTEN WITH A DIFFERANT DATA PATTERN
                ;2975 ; THAN THE LOCATION WAS VALIDATED WITH. THE REVERSE PARITY IS THEN
                ;2976 ; DISABLED AND THE LOCATION IS READ TO SEE THAT THE DATA CHANGED.
                ;2977 ; THE LOCATION WILL STILL BE A HIT SINCE THE REVERSE PARITY FUNCTION
                ;2978 ; DISABLES THE WRITE PULSE TO THE ADDRESS MATRIX BUT NOT THE DATA MATRIX.
                ;2979 ;
                ;2980 ; NOTE: ADDRESS 0 IS USED FOR THE GROUP C TEST SINCE THE VALID BIT
                ;2981 ;       IS USED IN THE GROUP C PARITY CHECK.
                ;2982 ;
                ;2983 ; LOGIC DESCRIPTION
                ;2984 ;
                ;2985 ; THIS TEST CHECKS THE GATES ON THE CAM MODULE THAT GENERATE THE
                ;2986 ; CACHE WRITE PULSE FOR AN "INVALIDATE" CYCLE AND A CACHE ADDRESS
                ;2987 ; PARITY ERROR.
                ;2988 ;
                ;2989 ; ERROR DESCRIPTION
                ;2990 ;
                ;2991 ; DATA: EXPECTED READ DATA
                ;2992 ;       RECEIVED READ DATA
                ;2993 ;       LOOP COUNT - INDICATES WHICH BYTE WAS BEING FORCED TO HAVE
                ;2994 ;    BAD PARITY. I.E. 3=BYTE A, 2=BYTE B, 1=BYTE C
                ;2995 ;-
                ;2996 ;*************************************************************
                ;2997 =0
U 1090, 0018,0039,0D80,09F8,0000,105B  ;2998 T001: NEWTST[.3]
U 1091, 0000,003C,0180,0800,0000,1092  ;2999  NOP
U 1092, 0000,003D,0180,0800,0000,104C  ;3000 =0 CALL[MAINTREGSETUP]  ; Set up Constants
U 1093, 0000,003C,8580,0800,0084,7252  ;3001  SC_K[.C]   ; INIT THE ADDRESS FOR VALIDATION
U 1252, 0003,001C,0180,0980,0000,1253  ;3002  RC[0]_NOT.MASK   ; ...
U 1253, 0000,003C,6580,0800,0084,7254  ;3003  SC_K[.10]   ; INIT THE REVERSE PARITY AND
U 1254, 0818,0038,A5F8,0800,0000,1255  ;3004  D_K[.60],Q_0   ; SBI INVALIDATE BITS IN THE
U 1255, 0819,0014,0180,0800,0000,1256  ;3005  D_D+K[.8]   ; MAINTENANCE REGISTER
U 1256, 0D00,003C,0180,0800,0000,1257  ;3006  D_DAL.SC
U 1257, 0001,003C,0180,0988,0000,1258  ;3007  RC[1]_D   ; SAVE
U 1258, 0018,0038,0D80,09E0,0000,1259  ;3008  RC[0C]_K[.3]   ; SET THE LOOP COUNT
U 1259, 0000,003C,8D80,0800,0084,725A  ;3009  SC_K[.1F]   ; GENERATE EXPECTED DATA
U 125A, 0000,003C,0980,0800,0084,B25B  ;3010  SC_SC-K[.2]
U 125B, 0003,001C,0180,09F0,0000,1094  ;3011  RC[0E]_NOT.MASK   ; SAVE
                ;3012
U 1094, 0000,003D,0180,0800,0000,1163  ;3013 =0 ERLOOP
```

```
U 1095. 0010.0038.0180.0900.0200.125C  ;3014 LOOP1: VA_RC[0]    ; LOAD ADDRESS TO INVALIDATE
U 125C. 0818.0038.0580.0800.0000.125D  ;3015  D_K[.1]     ; GET DATA PATTERN TO WRITE
U 125D. 0000.003C.0180.9800.0000.125E  ;3016  MCT/VALIDATE    ; VALIDATE THE LOCATION
U 125E. 0810.0038.75F0.2D08.0000.125F  ;3017  Q_ID[MAINT],D_RC[01]   ; GET CURRENT VALUE OF MAINTENANCE REG
U 125F. 081D.0030.0180.0800.0000.1260  ;3018  D_D.OR.Q    ; COMBINE WITH REVERSE PARITY BITS
U 1260. 0000.003C.7580.3C00.0000.1261  ;3019  ID[MAINT]_D    ; SET REVERSE PARITY BITS
U 1261. 0F03.003C.0180.0800.0200.1262  ;3020  VA_ALU,ALU_0(A),D_0   ; LOAD ADDRESS 0
U 1262. 0000.003C.0180.A800.0000.1263  ;3021  CACHE.P_D[LONG]    ; WRITE LOCATION ZERO
U 1263. 0000.003C.0180.0800.0000.1264  ;3022  NOP     ; WAIT FOR WRITE TO COMPLETE
U 1264. 0000.003C.0180.0800.0000.1265  ;3023  NOP
U 1265. 0000.003C.0180.0800.0000.1266  ;3024  NOP
U 1266. 0000.003C.0180.0800.0000.1267  ;3025  NOP
U 1267. 0000.003C.0180.0800.0000.1268  ;3026  NOP
U 1268. 0000.003C.0180.0800.0000.1269  ;3027  NOP
U 1269. 0000.003C.0180.0800.0000.126A  ;3028  NOP
U 126A. 0000.003C.0180.0800.0000.126B  ;3029  NOP
U 126B. 0000.003C.0180.0800.0000.126C  ;3030  NOP
U 126C. 0000.003C.0180.0800.0000.126D  ;3031  NOP
U 126D. 0800.003C.0180.0A30.0000.126E  ;3032  D_R[6]     ; GET DATA TO CLEAR REVERSE PARITY
U 126E. 0010.0038.7580.3D00.0200.126F  ;3033  VA_RC[0],ID[MAINT]_D   ; LOAD ADDRESS AND CLEAR REVERSE PARITY BITS
U 126F. 0000.003C.0180.0800.0000.1270  ;3034  NOP
U 1270. 0000.003C.0180.C800.0000.1271  ;3035  D[LONG]_CACHE.P    ; READ THE LOCATION
U 1271. 0010.0038.01C0.0970.0000.1272  ;3036  Q_RC[0E]    ; GET EXPECTED DATA
U 1272. 001D.2000.0180.0800.0010.1273  ;3037  ALU_Q-D,CLK.UBCC   ; CHECK
U 1273. 0000.013C.0180.0800.0000.1096  ;3038  Z?
U 1096. 0001.003D.0180.09E8.0000.1166  ;3039 =0 ERROR2,RC[0D]_D    ; DATA WAS NOT WRITTEN AS ZERO
U 1097. 0810.0038.D5F8.0900.0084.7274  ;3040  D_RC[0],SC_K[.6],Q_0   ; SELECT THE NEXT ADDRESS
U 1274. 0D00.003C.0180.0800.0000.1275  ;3041  D_DAL.SC    ; ...
U 1275. 0001.003C.0180.0980.0000.1276  ;3042  RC[0]_D     ; SAVE
U 1276. 0818.0038.4580.0800.0000.1277  ;3043  D_K[.8000]    ; GENERATE INCREMENT FOR PARITY BITS
U 1277. 0110.0038.01C0.0908.0000.1278  ;3044  D_D.LEFT2,Q_RC[1]   ; ...
U 1278. 001D.0014.0180.0988.0000.1279  ;3045  RC[1]_D+Q    ; UPDATE PARITY REVERSE BITS
U 1279. 0810.0038.0180.0960.0000.127A  ;3046  D_RC[0C]    ; GET THE LOOP COUNT
U 127A. 0019.0000.0980.0800.0010.127B  ;3047  ALU_D-K[.2],CLK.UBCC   ; SECOND TIME THRU LOOP?
U 127B. 0000.013C.0180.0800.0000.1098  ;3048  Z?     ; BRANCH IF YES
U 1098. 0000.003C.0180.0800.0000.127C  ;3049 =0 J/CONTINUE
U 1099. 0003.003C.0180.0980.0000.127C  ;3050  RC[0]_0     ; SET THE TEST ADDRESS TO ZERO
            ;3051 CONTINUE:
U 127C. 0019.0000.0580.09E0.0010.127D  ;3052  RC[0C]_D-K[.1],CLK.UBCC   ; DECREMENT AND CHECK
U 127D. 0000.013C.0180.0800.0000.109A  ;3053  Z?
U 109A. 0000.003C.0180.0800.0000.1095  ;3054 =0 J/LOOP1     ; CONTINUE TEST
U 109B. 0000.003C.0180.0800.0000.109C  ;3055 END001: NOP
            ;3056
            ;3057
            ;3058
```

```
          ;3059 .PAGE "TEST 206 SBI INVALIDATE WITH GROUP 1 ADDRESS PARITY ERROR"
          ;3060 ;*************************************************************************
          ;3061 ;*
          ;3062 ;
          ;3063 ; SBI INVALIDATE WITH GROUP 1 ADDRESS PARITY ERROR
          ;3064 ;
          ;3065 ; TEST DESCRIPTION
          ;3066 ;
          ;3067 ; THIS TEST USES THE MAINTENANCE SBI INVALIDATE FUNCTION TO CHECK
          ;3068 ; THAT THE CACHE WRITE PULSE IS GENERATED IF THE LOCATION TO BE
          ;3069 ; INVALIDATED CONTAINS A PARITY ERROR.
          ;3070 ;
          ;3071 ; THIS IS DONE WITH THREE CACHE ADDRESSES (1000, 40000, AND 0)
          ;3072 ; TO CHECK A PARITY ERROR IN EACH BYTE. THE ADDRESS IS FIRST WRITTEN
          ;3073 ; VALID, THE REVERSE PARITY BITS AND SBI INVALIDATE BITS
          ;3074 ; ARE SET, AND LOCATION 0 IS WRITTEN WITH A DIFFERANT DATA PATTERN
          ;3075 ; THAN THE LOCATION WAS VALIDATED WITH. THE REVERSE PARITY IS THEN
          ;3076 ; DISABLED AND THE LOCATION IS READ TO SEE THAT THE DATA CHANGED.
          ;3077 ; THE LOCATION WILL STILL BE A HIT SINCE THE REVERSE PARITY FUNCTION
          ;3078 ; DISABLES THE WRITE PULSE TO THE ADDRESS MATRIX BUT NOT THE DATA MATRIX.
          ;3079 ;
          ;3080 ; NOTE: ADDRESS 0 IS USED FOR THE GROUP C TEST SINCE THE VALID BIT
          ;3081 ;  IS USED IN THE GROUP C PARITY CHECK.
          ;3082 ;
          ;3083 ; LOGIC DESCRIPTION
          ;3084 ;
          ;3085 ; THIS TEST CHECKS THE GATES ON THE CAM MODULE THAT GENERATE THE
          ;3086 ; CACHE WRITE PULSE FOR AN "INVALIDATE" CYCLE AND A CACHE ADDRESS
          ;3087 ; PARITY ERROR.
          ;3088 ;
          ;3089 ; ERROR DESCRIPTION
          ;3090 ;
          ;3091 ; DATA: EXPECTED READ DATA
          ;3092 ;       RECEIVED READ DATA
          ;3093 ;       LOOP COUNT - INDICATES WHICH BYTE WAS BEING FORCED TO HAVE
          ;3094 ;    BAD PARITY. I.E. 3=BYTE A, 2=BYTE B, 1=BYTE C
          ;3095 ;-
          ;3096 ;*************************************************************************
          ;3097 =0
U 109C, 0018.0039.0D80.09F8.0000.105B  ;3098 T002: NEWTST[.3]
U 109D, 0000.003C.0180.0800.0000.109E  ;3099  NOP
U 109E, 0000.003D.0180.0800.0000.104C  ;3100 =0 CALL[MAINTREGSETUP]  ; SETUP SOME REGISTERS
U 109F, 0818.0038.4580.0800.0000.127E  ;3101  D_K[.8000]   ; SETUP TO FORCE REPLACE G1 AND
U 127E, 0500.003C.8980.0800.0084.727F  ;3102  D_D.LEFT,SC_K[.D]  ; FORCE MISS GROUP 0
U 127F, 0801.001C.0180.0800.0000.1280  ;3103  D_D.ORNOT.MASK
U 1280, 0001.003C.7580.3EA0.0000.1281  ;3104  R[4]_D,ID[MAINT]_D  ; ...
U 1281, 0000.003C.8580.0800.0084.7282  ;3105  SC_K[.C]   ; INIT THE ADDRESS FOR VALIDATION
U 1282, 0003.001C.0180.0980.0000.1283  ;3106 = RC[0]_NOT.MASK   ; ...
U 1283, 0000.003C.6580.0800.0084.7284  ;3107  SC_K[.10]   ; INIT THE REVERSE PARITY AND
U 1284, 0818.0038.A5F8.0800.0000.1285  ;3108  D_K[.60],Q_0   ; SBI INVALIDATE BITS IN THE
U 1285, 0819.0014.0980.0800.0000.1286  ;3109  D_D+K[.2]   ; MAINTENANCE REGISTER
U 1286, 0D00.003C.0180.0800.0000.1287  ;3110  D_DAL.SC
U 1287, 0001.003C.0180.0980.0000.1288  ;3111  RC[1]_D    ; SAVE
U 1288, 0018.0038.0D80.09E0.0000.1289  ;3112  RC[0C]_K[.3]   ; SET THE LOOP COUNT
U 1289, 0000.003C.8D80.0800.0084.728A  ;3113  SC_K[.1F]   ; GENERATE THE EXPECTED DATA
```

```
U 128A, 0000.003C.0980.0800.0084.B28B  ;3114  SC_SC-K[.2]
U 128B, 0003.001C.0180.09F0.0000.10A0  ;3115  RC[0E]_NOT.MASK
        ;3116
U 10A0, 0000.003D.0180.0800.0000.1163  ;3117 =0 ERLOOP
U 10A1, 0010.0038.0180.0900.0200.128C  ;3118 LOOP2: VA_RC[0]   ; LOAD ADDRESS TO INVALIDATE
U 128C, 0818.0038.0580.0800.0000.128D  ;3119  D_K[.1]    ; GET DATA PATTERN TO WRITE
U 128D, 0000.003C.0180.9800.0000.128E  ;3120  MCT/VALIDATE   ; VALIDATE THE LOCATION
U 128E, 0810.0038.75F0.2D08.0000.128F  ;3121  Q_ID[MAINT],D_RC[01]  ; GET CURRENT VALUE OF MAINTENANCE REG
U 128F, 081D.0030.0180.0800.0000.1290  ;3122  D_D.OR.Q   ; COMBINE WITH REVERSE PARITY BITS
U 1290, 0000.003C.7580.3C00.0000.1291  ;3123  ID[MAINT]_D   ; SET REVERSE PARITY BITS
U 1291, 0F03.003C.0180.0800.0200.1292  ;3124  VA_ALU,ALU_0(A),D_0  ; LOAD ADDRESS 0
U 1292, 0000.003C.0180.A800.0000.1293  ;3125  CACHE.P_D[LONG]   ; WRITE LOCATION ZERO
U 1293, 0000.003C.0180.0800.0000.1294  ;3126  NOP    ; WAIT FOR WRITE TO COMPLETE
U 1294, 0000.003C.0180.0800.0000.1295  ;3127  NOP
U 1295, 0000.003C.0180.0800.0000.1296  ;3128  NOP
U 1296, 0000.003C.0180.0800.0000.1297  ;3129  NOP
U 1297, 0000.003C.0180.0800.0000.1298  ;3130  NOP
U 1298, 0000.003C.0180.0800.0000.1299  ;3131  NOP
U 1299, 0000.003C.0180.0800.0000.129A  ;3132  NOP
U 129A, 0000.003C.0180.0800.0000.129B  ;3133  NOP
U 129B, 0000.003C.0180.0800.0000.129C  ;3134  NOP
U 129C, 0000.003C.0180.0800.0000.129D  ;3135  NOP
U 129D, 0800.003C.0180.0A20.0000.129E  ;3136  D_R[4]    ; GET DATA TO CLEAR REVERSE PARITY
U 129E, 0010.0038.7580.3D00.0200.129F  ;3137  VA_RC[0],ID[MAINT]_D  ; LOAD ADDRESS AND CLEAR REVERSE PARITY BITS
U 129F, 0000.003C.0180.0800.0000.12A0  ;3138  NOP
U 12A0, 0000.003C.0180.C800.0000.12A1  ;3139  D[LONG]_CACHE.P   ; READ THE LOCATION
U 12A1, 0010.0038.01C0.0970.0000.12A2  ;3140  Q_RC[0E]   ; GET EXPECTED DATA
U 12A2, 001D.2000.0180.0800.0010.12A3  ;3141  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 12A3, 0000.013C.0180.0800.0000.10A2  ;3142  Z?
U 10A2, 0001.003D.0180.09E8.0000.1166  ;3143 =0 ERROR2,RC[0D]_D   ; DATA WAS NOT WRITTEN AS ZERO
U 10A3, 0810.0038.D5F8.0900.0084.72A4  ;3144  D_RC[0],SC_K[.6],Q_0  ; SELECT THE NEXT ADDRESS
U 12A4, 0D00.003C.0180.0800.0000.12A5  ;3145  D_DAL.SC   ; ...
U 12A5, 0001.003C.0180.0980.0000.12A6  ;3146  RC[0]_D    ; SAVE
U 12A6, 0818.0038.4580.0800.0000.12A7  ;3147  D_K[.8000]   ; GENERATE INCREMENT FOR PARITY BITS
U 12A7, 0110.0038.01C0.0908.0000.12A8  ;3148  D_D.LEFT2,Q_RC[1]  ; ...
U 12A8, 001D.0014.0180.0988.0000.12A9  ;3149  RC[1]_D+Q   ; UPDATE PARITY REVERSE BITS
U 12A9, 0810.0038.0180.0960.0000.12AB  ;3150  D_RC[0C]   ; GET THE LOOP COUNT
U 12AB, 0019.0000.0980.0800.0010.12AC  ;3151  ALU_D-K[.2],CLK.UBCC  ; SECOND TIME THRU THE LOOP?
U 12AC, 0000.013C.0180.0800.0000.10A4  ;3152  Z?    ; BRANCH IF YES
U 10A4, 0000.003C.0180.0800.0000.12AD  ;3153 =0 J/CONTINUE1
U 10A5, 0003.003C.0180.0980.0000.12AD  ;3154  RC[0]_0
        ;3155 CONTINUE1:
U 12AD, 0019.0000.0580.09E0.0010.12AE  ;3156  RC[0C]_D-K[.1],CLK.UBCC  ; DECREMENT AND CHECK
U 12AE, 0000.013C.0180.0800.0000.10A6  ;3157  Z?
U 10A6, 0000.003C.0180.0800.0000.10A1  ;3158 =0 J/LOOP2    ; CONTINUE TEST
U 10A7, 0000.003C.0180.0800.0000.10A8  ;3159 END002: NOP
        ;3160
        ;3161
        ;3162
```

```
;3163 .PAGE "TEST 207 CPU READ AND SBI INVALIDATE SIMULTANEOUSLY"
;3164 ;*************************************************************************
;3165 ;*
;3166 ;
;3167 ; CPU READ AND SBI INVALIDATE SIMULTANEOUSLY"
;3168 ;
;3169 ; TEST DESCRIPTION
;3170 ;
;3171 ; THIS TEST CHECKS THE SBI INVALIDATE FUNCTION USING THE UNIBUS
;3172 ; ADAPTER TO GENERATE THE WRITE TO MEMORY. THE TEST IS STRUCTURED
;3173 ; TO CHECK THE CASE WHERE THE CPU IS WAITING FOR READ DATA AND THE
;3174 ; UBA ISSUES A WRITE COMMAND TO THE MEMORY. WHEN THE CPU RECEIVES
;3175 ; THE READ DATA, IT SHOULD BE MARKED INVALID IN THE CACHE IF THE
;3176 ; UBA REFERANCED THE SAME (ALMOST) LOCATION IS THE CPU.
;3177 ; DIFFERANT ADDRESS PATTERNS ARE USED TO CHECK THE ADDRESS COMPARATOR
;3178 ; ON THE SBL MODULE.
;3179 ;
;3180 ; SUBTEST 1 - CHECK THE MARK INVALID COMPARATOR WITH ALL ZERO'S AND
;3181 ;      ALL ONE'S. SHOULD INVALIDATE THE READ DATA.
;3182 ;
;3183 ; SUBTEST 2 - SET THE A LEG AT ALL ZERO'S AND FLOAT A ONE ACROSS THE
;3184 ;      B LEG OF THE MARK INVALID COMPARATOR.
;3185 ;
;3186 ; SUBTEST 3 - SET THE B LEG AT ALL ZERO'S AND FLOAT A ONE ACROSS THE
;3187 ;      A LEG OF THE MARK INVALID COMPARATOR.
;3188 ;
;3189 ; SUBTEST 4 - SET THE A LEG TO ALL ONE'S AND FLOAT A ZERO ACROSS THE
;3190 ;      B LEG OF THE MARK INVALID COMPARATOR.
;3191 ;
;3192 ; SUBTEST 5 - SET THE B LEG TO ALL ONE'S AND FLOAT A ZERO ACROSS THE
;3193 ;      A LEG OF THE MARK INVALID COMPARATOR.
;3194 ;
;3195 ; LOGIC DESCRIPTION
;3196 ;
;3197 ; THIS TEST CHECKS THE ADDRESS COMPARATOR AND THE "MARK INVALID"
;3198 ; LATCH ON THE SBL MODULE.
;3199 ;
;3200 ; ERROR DESCRIPTION
;3201 ;
;3202 ; DATA: EXPECTED HIT AND MISS BITS
;3203 ;       RECEIVED HIT AND MISS BITS
;3204 ;       ADDRESS REFERANCED BY UBA (A LEG OF COMPARATOR)
;3205 ;       ADDRESS REFERANCE BY CPU (B LEG OF COMPARATOR)
;3206 ;
;3207 ; NOTE: THE UBA ADDRESS IS ADJUSTED TO THE SAME BIT POSITIONS AS
;3208 ;  THE CPU ADDRESS.
;3209 ;-
;3210 ;*************************************************************************
;3211
;3212 =0
U 10A8, 0018,0039,1180,09F8,0000,105B  ;3213 T003: NEWTST[.4]
U 10A9, 0000,003C,0180,0800,0000,10AA  ;3214  NOP
U 10AA, 0000,003D,0180,0800,0000,104C  ;3215 =0 CALL,J/MAINTREGSETUP
U 10AB, 0000,003C,0180,0800,0000,10AC  ;3216  NOP
U 10AC, 0000,003D,0180,0800,0000,1213  ;3217 =0 CALL,J/LOOKFORDEVICE  ; TRY AND FIND A UBA
```

```
U 10AD. 0000.003C.0180.0A00.0210.12AF  ;3218 CHECKR0:VA_ALU,ALU_R[0],CLK.UBCC ; WAS THERE A DEVICE?
U 12AF. 0000.013C.0180.0800.0000.10AE  ;3219  Z?     ; BRANCH IF NO
U 10AE. 0000.003C.0180.0800.0000.12B0  ;3220 =0 J/CHECKCODE1    ; CHECK IF ITS A UBA
U 10AF. 0000.003C.0180.0800.0000.10F2  ;3221  J/END003    ; NO UBA AVAILABLE
        ;3222
        ;3223 CHECKCODE1:
U 12B0. 0000.003C.0180.C800.0000.12B1  ;3224  D[LONG]_CACHE.P    ; READ THE CONFIG REG
U 12B1. 0200.003C.0180.0800.0000.12B2  ;3225  D_D.RIGHT2    ; CHECK BITS 3 AND 4
U 12B2. 0600.003C.0180.0800.0000.12B3  ;3226  D_D.RIGHT    ; ...
U 12B3. 0000.193C.0180.0800.0000.103C  ;3227  D1-0?     ; UBA = 01
U 103C. 0000.003C.0180.0800.0000.12BC  ;3228 =1100 J/LOOKAGAIN    ; THIS IS MBA
U 103D. 0000.003C.0180.C800.0000.12B4  ;3229  D[LONG]_CACHE.P    ; GET CONFIG REG AGAIN
U 12B4. 0819.0034.0D80.0800.0000.12B5  ;3230 = D_D.AND.K[.3]    ; GENERATE THE BASE ADDRESS
U 12B5. 0819.0030.1180.0800.0000.12B6  ;3231  D_D.OR.K[.4]    ; OF UNIBUS SPACE
U 12B6. 0000.003C.21F8.0800.0084.72B7  ;3232  SC_K[.14],Q_0
U 12B7. 0000.003C.0980.0800.0084.B2B8  ;3233  SC_SC-K[.2]
U 12B8. 0D00.003C.D980.0800.0000.12B9  ;3234  D_DAL.SC,K[.9]
U 12B9. 0000.003C.D980.0800.0084.92BA  ;3235  SC_SC+K[.9]
U 12BA. 0000.003C.0980.0800.0084.92BB  ;3236  SC_SC+K[.2]
U 12BB. 0001.001C.0180.0980.0000.10B2  ;3237  RC[0]_ALU,ALU_D.ORNOT.MASK,J/START1    ; ...
        ;3238
        ;3239 LOOKAGAIN:
U 12BC. 0000.003C.0180.0800.0084.72BD  ;3240  SC_K[.8]
U 12BD. 0818.0038.75F8.0800.0000.12BE  ;3241  D_K[.20],Q_0
U 12BE. 0D00.003C.0180.0800.0000.10B0  ;3242  D_DAL.SC
U 10B0. 0000.003D.89E0.0800.0084.708B  ;3243 =0 Q_D,SC_K[.D],CALL,J/FOUNDMEMORY
U 10B1. 0000.003C.0180.0800.0000.10AD  ;3244  J/CHECKR0
        ;3245
        ;3246 =0
U 10B2. 0000.003D.0180.0800.0000.123F  ;3247 START1: CALL,J/WAITFORUNIBUS
U 10B3. 0000.003C.75F0.2C00.0000.12BF  ;3248  Q_ID[MAINT]    ; SET THE SBI INVALIDATE ENABLE BIT
U 12BF. 0000.003C.2180.0800.0084.72C0  ;3249  SC_K[.14]
U 12C0. 0000.003C.0580.0800.0084.92C1  ;3250  SC_SC+K[.1]
U 12C1. 0801.201C.0180.0800.0000.12C2  ;3251  D_Q.ORNOT.MASK
U 12C2. 0000.003C.7580.3C00.0000.12C3  ;3252  ID[MAINT]_D    ; ...
U 12C3. 0800.003C.F580.0A00.0084.72C4  ;3253  D_R[0],SC_K[.A]    ; GENERATE ADDRESS OF THE UBA
U 12C4. 0000.003C.0580.0800.0084.92C5  ;3254  SC_SC+K[.1]
U 12C5. 0001.001C.0180.0988.0000.12C6  ;3255  RC[1]_D.ORNOT.MASK    ; SAVE IN RC[1] = 200XX800
U 12C6. 0800.003C.0180.0A00.0000.12C7  ;3256  D_R[0]     ; GET BASE ADDRESS AGAIN
U 12C7. 0819.0030.3180.0800.0000.12C8  ;3257  D_D.OR.K[.40]    ; GENERATE ADDRESS OF BUFFERED DATA
U 12C8. 0019.0030.1180.0A98.0000.12C9  ;3258  R[3]_ALU,ALU_D.OR.K[.4]    ; PATH REGISTER 1
U 12C9. 0818.0038.4580.0800.0000.12CA  ;3259  D_K[.8000]    ; GENERATE BASE CONTENTS OF MAP
U 12CA. 0819.0030.7580.0800.0000.12CB  ;3260  D_D.OR.K[.20]
U 12CB. 0000.003C.65F8.0800.0084.72CC  ;3261  SC_K[.10],Q_0
U 12CC. 0D00.003C.0180.0800.0000.12CD  ;3262  D_DAL.SC
U 12CD. 0001.003C.0180.0AA8.0000.10B4  ;3263  R[5]_D     ; SAVE IN R5 = 80200000
        ;3264
        ;3265 ;////////////////////////////////////////////////////////////////////////
        ;3266 ;
        ;3267 ; FIRST CHECK FOR THE INVALIDATE WITH ZERO AND ONE'S ON THE LEGS.
        ;3268 ;
        ;3269
U 10B4. 0000.003D.0180.0800.0000.117D  ;3270 =0 SUBTEST
U 10B5. 0000.003C.0180.0A28.0000.12CE  ;3271  LAB_R[5]    ; GET CONTENTS OF MAP
U 12CE. 0000.003C.0180.0990.0000.12CF  ;3272  RC[2]_LA    ; SAVE
```

```
U 12CF. 0003.003C.0180.09D8.0000.12D0  ;3273  RC[0B]_0   ; AND THE CPU ADDRESS
U 12D0. 0003.003C.0180.09E0.0000.12D1  ;3274  RC[0C]_0   ; AND THE UBA ADDRESS
U 12D1. 0003.003C.0180.09F0.0000.12D2  ;3275  RC[0E]_0   ; AND EXPECTED HIT/MISS BITS
U 12D2. 0018.0038.0980.0998.0000.10B6  ;3276  RC[3]_K[.2]   ; SET THE LOOP COUNT
U 10B6. 0000.003D.0180.0800.0000.1163  ;3277 =0 ERLOOP
U 10B7. 0010.0038.0180.0908.0200.12D3  ;3278 LOOP3: VA_ALU,ALU_RC[1]  ; LOAD THE MAP ADDRESS
U 12D3. 0810.0038.0180.0910.0000.12D4  ;3279  D_RC[2]   ; GET CONTENTS OF MAP
U 12D4. 0000.003C.0180.A800.0000.12D5  ;3280  CACHE.P_D[LONG]   ; LOAD THE MAP
          ;3281
U 12D5. 0010.0038.0180.0900.0200.12D6  ;3282  VA_ALU,ALU_RC[0]  ; LOAD THE UNIBUS ADDRESS
U 12D6. 0000.403C.0180.A800.0000.12D7  ;3283  CACHE.P_D[WORD]   ; LOAD THE BUFFERED DATA PATH
U 12D7. 0010.0038.0180.0958.0200.12D8  ;3284  VA_ALU,ALU_RC[0B]  ; LOAD CPU ADDRESS
U 12D8. 0000.003C.0180.9000.0000.12D9  ;3285  CACHE.INVALIDATE  ; INVALIDATE THE CPU ADDRESS
U 12D9. 0000.003C.8D80.0A18.0284.72DA  ;3286  SC_K[.1F],VA_ALU,ALU_R[3] ; GENERATE DATA TO PURGE DATA PATH
U 12DA. 0818.0038.6580.0800.0000.12DB  ;3287  D_K[.10]   ; WAIT FOR BDP TO FINISH
U 12DB. 0819.0000.0580.0800.0010.12DC  ;3288 BDPW: D_D-K[.1],CLK.UBCC  ; LOADING
U 12DC. 0000.013C.0180.0800.0000.10B8  ;3289  Z?
U 10B8. 0000.003C.0180.0800.0000.12DB  ;3290 =0 J/BDPW    ; WAIT SOME MORE
U 10B9. 0803.001C.0180.0800.0000.12DD  ;3291  D_NOT.MASK
U 12DD. 0000.003C.0180.A800.0000.12DE  ;3292  CACHE.P_D[LONG]   ; PURGE THE DATA PATH
U 12DE. 0000.003C.0180.0800.0000.12DF  ;3293  NOP
U 12DF. 0000.003C.0180.0800.0000.12E0  ;3294  NOP
U 12E0. 0000.003C.0180.0800.0000.12E1  ;3295  NOP
U 12E1. 0000.003C.0180.0800.0000.12E2  ;3296  NOP
U 12E2. 0000.003C.0180.0800.0000.12E3  ;3297  NOP
U 12E3. 0010.0038.0180.0958.0200.12E4  ;3298  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 12E4. 0000.003C.0180.C800.0000.12E5  ;3299  D[LONG]_CACHE.P   ; START THE CPU READ
          ;3300      ; UBA SHOULD ISSUE C/A TO THE SBI
          ;3301      ; MEMORY DURING THIS READ
          ;3302
U 12E5. 0000.003C.0180.C800.0000.12E6  ;3303  D[LONG]_CACHE.P   ; THIS READ SHOULD STILL BE A MISS
U 12E6. 0000.003C.75F0.2C00.0000.12E7  ;3304  Q_ID[MAINT]   ; READ THE HIT/MISS BITS
U 12E7. 001C.0034.01C0.0A38.0010.12E8  ;3305  Q_Q.AND.R[7],CLK.UBCC  ; MASK THE HIT/MISS BITS
U 12E8. 0000.013C.0180.0800.0000.10BA  ;3306  Z?    ; CHECK THAT IT WAS A MISS
U 10BA. 0001.203D.0180.09E8.0000.1166  ;3307 =0 ERROR2,RC[0D]_Q   ; READ DATA WAS NOT INVALIDATED
          ;3308
U 10BB. 0000.003C.0180.0800.0000.10BC  ;3309  NOP
U 10BC. 0818.0039.8D80.0800.0000.1170  ;3310 =0 D_K[.1F],CALL,J/DC8  ; GENERATE CONTENTS OF MAP REGISTER
U 10BD. 0010.0038.01C0.0910.0000.12E9  ;3311  Q_RC[2]
U 12E9. 001D.0014.D9F8.0990.0084.72EA  ;3312  RC[2]_D+Q,SC_K[.9],Q_0  ; PUT IN RC[2]
U 12EA. 0D00.003C.0180.0800.0000.12EB  ;3313  D_DAL.SC   ; GENERATE DATA FOR TYPEOUT
U 12EB. 0001.003C.0180.09E0.0000.12EC  ;3314  RC[0C]_D    ; SAVE
U 12EC. 0818.0038.5580.0800.0000.12ED  ;3315  D_K[.3F]   ; GENERATE CPU ADDRESS
U 12ED. 0000.003C.8580.0800.0084.72EE  ;3316  SC_K[.C]   ; ...
U 12EE. 0D00.003C.0180.0800.0000.12EF  ;3317  D_DAL.SC
U 12EF. 0001.003C.0180.09D8.0000.12F0  ;3318  RC[0B]_D   ; SAVE = 3F000
U 12F0. 0810.0038.0180.0918.0000.12F1  ;3319  D_RC[3]    ; GET THE LOOP COUNT
U 12F1. 0019.0000.0580.0998.0010.12F2  ;3320  RC[3]_D-K[.1],CLK.UBCC  ; DECREMENT AND CHECK
U 12F2. 0000.013C.0180.0800.0000.10BE  ;3321  Z?
U 10BE. 0000.003C.0180.0800.0000.10B7  ;3322 =0 J/LOOP3    ; NOT FINISHED
U 10BF. 0000.003C.0180.0800.0000.10C0  ;3323  NOP
          ;3324
          ;3325 ;////////////////////////////////////////////////////////////////////
          ;3326 ;*
          ;3327 ; NOW SET THE ALEG AT ZERO AND FLOAT A ONE THRU THE B LEG OF THE COMPARATOR.
```

```
            ;3328 ;-
            ;3329
U 10C0. 0000.003D.0180.0800.0000.117D  ;3330 =0 SUBTEST
U 10C1. 0003.003C.0180.09E0.0000.12F3  ;3331  RC[0C]_0    ; INIT UBA ADDRESS FOR TYPEOUT
U 12F3. 0000.003C.0180.0A28.0000.12F4  ;3332  LAB_R[5]    ; GET CONTENTS OF MAP
U 12F4. 0000.003C.0180.0990.0000.12F5  ;3333  RC[2]_LA     ; SAVE
U 12F5. 0000.003C.8580.0800.0084.72F6  ;3334  SC_K[.C]
U 12F6. 0003.001C.0180.09D8.0000.12F7  ;3335  RC[0B]_NOT.MASK    ; INIT THE CPU ADDRESS
U 12F7. 0010.0038.0180.0908.0200.12F8  ;3336  VA_ALU,ALU_RC[1]  ; LOAD THE MAP ADDRESS
U 12F8. 0810.0038.0180.0910.0000.12F9  ;3337  D_RC[2]
U 12F9. 0000.003C.0180.A800.0000.12FA  ;3338  CACHE.P_D[LONG]    ; LOAD THE MAP
U 12FA. 0800.003C.0180.0A40.0000.12FB  ;3339  D_R[8]
U 12FB. 0001.003C.0180.09F0.0000.12FC  ;3340  RC[0E]_D    ; SET THE EXPECTED HIT/MISS BITS
U 12FC. 0018.0038.D580.0998.0000.10C2  ;3341  RC[3]_K[.6]    ; SET THE LOOP COUNT
U 10C2. 0000.003D.0180.0800.0000.1163  ;3342 =0 ERLOOP
U 10C3. 0010.0038.0180.0900.0200.12FD  ;3343 LOOP4: VA_ALU,ALU_RC[0]  ; LOAD THE UNIBUS ADDRESS
U 12FD. 0000.403C.0180.A800.0000.12FE  ;3344  CACHE.P_D[WORD]    ; LOAD THE BUFFERED DATA PATH
U 12FE. 0010.0038.0180.0958.0200.12FF  ;3345  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 12FF. 0000.003C.0180.9000.0000.1300  ;3346  CACHE.INVALIDATE  ; INVALIDATE THE CACHE
U 1300. 0000.003C.8D80.0A18.0284.7301  ;3347  VA_ALU,ALU_R[3],SC_K[.1F] ; LOAD ADDRESS OF DPR 1
U 1301. 0818.0038.6580.0800.0000.1302  ;3348  D_K[.10]    ; WAIT FOR BDP TO FINISH
U 1302. 0819.0000.0580.0800.0010.1303  ;3349 BDPW1: D_D-K[.1],CLK.UBCC  ; LOADING
U 1303. 0000.013C.0180.0800.0000.10C4  ;3350  Z?
U 10C4. 0000.003C.0180.0800.0000.1302  ;3351 =0 J/BDPW1     ; WAIT SOME MORE
U 10C5. 0803.001C.0180.0800.0000.1304  ;3352  D_NOT.MASK
U 1304. 0000.003C.0180.A800.0000.1305  ;3353  CACHE.P_D[LONG]    ; START THE UBA
U 1305. 0000.003C.0180.0800.0000.1306  ;3354  NOP
U 1306. 0000.003C.0180.0800.0000.1307  ;3355  NOP
U 1307. 0000.003C.0180.0800.0000.1308  ;3356  NOP
U 1308. 0000.003C.0180.0800.0000.1309  ;3357  NOP
U 1309. 0000.003C.0180.0800.0000.130A  ;3358  NOP
U 130A. 0010.0038.0180.0958.0200.130B  ;3359  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 130B. 0000.003C.0180.C800.0000.130C  ;3360  D[LONG]_CACHE.P    ; START THE CPU READ
            ;3361
U 130C. 0000.003C.0180.C800.0000.130D  ;3362  D[LONG]_CACHE.P    ; THIS SHOULD BE A HIT
U 130D. 0800.003C.75F0.2E40.0000.130E  ;3363  Q_ID[MAINT],D_R[8]  ; READ THE HIT/MISS BITS
U 130E. 001C.0034.01C0.0A38.0000.130F  ;3364  Q_Q.AND.R[7]    ; MASK
U 130F. 001D.2000.0180.0800.0010.1310  ;3365  ALU_Q-D,CLK.UBCC  ; CHECK FOR A HIT
U 1310. 0000.013C.0180.0800.0000.10C6  ;3366  Z?
U 10C6. 0001.203D.0180.09E8.0000.1166  ;3367 =0 ERROR2,RC[0D]_Q    ; SECOND REFERANCE WAS NOT A HIT
            ;3368
U 10C7. 0810.0038.0180.0958.0000.1311  ;3369  D_RC[0B]    ; GET THE CPU ADDRESS
U 1311. 0500.003C.0180.0800.0000.1312  ;3370  D_D.LEFT    ; SELECT NEXT PATTERN
U 1312. 0001.003C.0180.09D8.0000.1313  ;3371  RC[0B]_D    ; SAVE
U 1313. 0810.0038.0180.0918.0000.1314  ;3372  D_RC[3]     ; GET THE LOOP COUNT
U 1314. 0019.0000.0580.0998.0010.1315  ;3373  RC[3]_D-K[.1],CLK.UBCC  ; DECREMENT AND CHECK
U 1315. 0000.013C.0180.0800.0000.10C8  ;3374  Z?
U 10C8. 0000.003C.0180.0800.0000.10C3  ;3375 =0 J/LOOP4
U 10C9. 0000.003C.0180.0800.0000.10CA  ;3376  NOP
            ;3377
            ;3378 ;////////////////////////////////////////////////////////////////////////
            ;3379 ;*
            ;3380 ; NOW PUT ZERO'S ON THE B LEG AND FLOAT A ONE ACROSS.THE A LEG
            ;3381 ;-
            ;3382
```

```
U 10CA, 0000.003D.0180.0800.0000.117D  ;3383 =0 SUBTEST
U 10CB, 0818.0038.0180.0800.0000.1316  ;3384  D_K[.8]     ; INIT TYPEOUT OF MAP CONTENTS
U 1316, 0000.003C.D9F8.0800.0084.7317  ;3385  SC_K[.9],Q_0
U 1317, 0D00.003C.0180.0800.0000.1318  ;3386  D_DAL.SC
U 1318, 0001.003C.0180.09E0.0000.1319  ;3387  RC[0C]_D    ; SAVE THE ADDRESS OF THE UBA FOR TYPEOUT
U 1319, 0800.003C.0180.0A28.0000.131A  ;3388  D_R[5]      ; GET BASE CONTENTS OF MAP
U 131A, 0819.0030.0180.0990.0000.131B  ;3389  D_D.OR.K[.8],RC[2]_ALU   ; SAVE = 80200008
U 131B, 0003.003C.0180.09D8.0000.131C  ;3390  RC[0B]_0    ; INIT THE ADDRESS OF THE CPU
U 131C, 0018.0038.D580.0998.0000.10CC  ;3391  RC[3]_K[.6]    ; SET THE LOOP COUNT
U 10CC, 0000.003D.0180.0800.0000.1163  ;3392 =0 ERLOOP
U 10CD, 0010.0038.0180.0958.0200.131D  ;3393 LOOP5: VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 131D, 0000.003C.0180.9000.0000.131E  ;3394  CACHE.INVALIDATE
U 131E, 0010.0038.0180.0908.0200.131F  ;3395  VA_ALU,ALU_RC[1]  ; LOAD THE ADDRESS OF THE MAP
U 131F, 0810.0038.0180.0910.0000.1320  ;3396  D_RC[2]    ; GET THE CONTENTS OF THE MAP
U 1320, 0000.003C.0180.A800.0000.1321  ;3397  CACHE.P_D[LONG]    ; LOAD THE MAP
U 1321, 0010.0038.0180.0900.0200.1322  ;3398  VA_ALU,ALU_RC[0]  ; LOAD THE ADDRESS OF THE UNIBUS
U 1322, 0000.403C.0180.A800.0000.1323  ;3399  CACHE.P_D[WORD]    ; LOAD THE BUFFERED DATA PATH
U 1323, 0000.003C.8D80.0A18.0284.7324  ;3400  VA_ALU,ALU_R[3],SC_K[.1F] ; LOAD ADDRESS OF DPR1
U 1324, 0818.0038.6580.0800.0000.1325  ;3401  D_K[.10]    ; WAIT FOR BDP TO FINISH
U 1325, 0819.0000.0580.0800.0010.1326  ;3402 BDPW2: D_D-K[.1],CLK.UBCC  ; LOADING
U 1326, 0000.013C.0180.0800.0000.10CE  ;3403  Z?
U 10CE, 0000.003C.0180.0800.0000.1325  ;3404 =0 J/BDPW2     ; WAIT SOME MORE
U 10CF, 0803.001C.0180.0800.0000.1327  ;3405  D_NOT.MASK
U 1327, 0000.003C.0180.A800.0000.1328  ;3406  CACHE.P_D[LONG]    ; START THE UBA
U 1328, 0000.003C.0180.0800.0000.1329  ;3407  NOP
U 1329, 0000.003C.0180.0800.0000.132A  ;3408  NOP
U 132A, 0000.003C.0180.0800.0000.132B  ;3409  NOP
U 132B, 0000.003C.0180.0800.0000.132C  ;3410  NOP
U 132C, 0000.003C.0180.0800.0000.132D  ;3411  NOP
U 132D, 0010.0038.0180.0958.0200.132E  ;3412  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 132E, 0000.003C.0180.C800.0000.132F  ;3413  D[LONG]_CACHE.P    ; START THE CPU READ
U 132F, 0000.003C.0180.C800.0000.1330  ;3414  D[LONG]_CACHE.P    ; THIS SHOULD BE A HIT
U 1330, 0810.0038.75F0.2D70.0000.1331  ;3415  Q_ID[MAINT],D_RC[0E]  ; READ THE HIT/MISS BITS
U 1331, 001C.0034.01C0.0A38.0000.1332  ;3416  Q_Q.AND.R[7]    ; MASK
U 1332, 001D.2000.0180.0800.0010.1333  ;3417  ALU_Q-D,CLK.UBCC  ; CHECK FOR A HIT
U 1333, 0000.013C.0180.0800.0000.10D0  ;3418  Z?
U 10D0, 0001.203D.0180.09E8.0000.1166  ;3419 =0 ERROR2,RC[0D]_Q    ; CPU ADDRESS IS NOT VALID
            ;3420
U 10D1, 0810.0038.01F8.0910.0000.1334  ;3421  D_RC[2],Q_0    ; GET THE CONTENTS OF THE MAP
U 1334, 0500.003C.D980.0800.0084.7335  ;3422  D_D.LEFT,SC_K[.9]
U 1335, 0D00.003C.0180.0800.0000.1336  ;3423  D_DAL.SC
U 1336, 0001.003C.0180.09E0.0000.1337  ;3424  RC[0C]_D
U 1337, 0810.0038.0180.0910.0000.1338  ;3425  D_RC[2]
U 1338, 0019.0034.C1C0.0800.0000.1339  ;3426  Q_D.AND.K[.FFFF]
U 1339, 001D.0014.0180.0990.0000.133A  ;3427  RC[2]_D+Q
U 133A, 0810.0038.0180.0918.0000.133B  ;3428  D_RC[3]     ; GET THE LOOP COUNT
U 133B, 0019.0000.0580.0998.0010.133C  ;3429  RC[3]_D-K[.1],CLK.UBCC
U 133C, 0000.013C.0180.0800.0000.10D2  ;3430  Z?
U 10D2, 0000.003C.0180.0800.0000.10CD  ;3431 =0 J/LOOP5
U 10D3, 0000.003C.0180.0800.0000.10D4  ;3432  NOP
            ;3433
            ;3434
            ;3435 ;////////////////////////////////////////////////////////////////////////
            ;3436 ;*
            ;3437 ; NOW SET THE A LEG TO ALL ONE'S AND FLOAT A ZERO ACROSS THE B LEG.
```

```
                ;3438 ;-
                ;3439
U 10D4. 0000.003D.0180.0800.0000.117D   ;3440 =0 SUBTEST
U 10D5. 0818.0038.8D80.0800.0000.133D   ;3441  D_K[.1F]    ; GENERATE THE INIT CPU ADDRESS
U 133D. 0000.003C.89F8.0800.0084.733E   ;3442  SC_K[.D],Q_0
U 133E. 0D00.003C.0180.0800.0000.133F   ;3443  D_DAL.SC
U 133F. 0001.003C.0180.09D8.0000.10D6   ;3444  RC[0B]_D    ; SAVE = 3E000
U 10D6. 0818.0039.8D80.0800.0000.1170   ;3445 =0 D_K[.1F],CALL,J/DC8  ; GENERATE INIT MAP CONTENTS
U 10D7. 081C.2030.01E0.0A28.0000.1340   ;3446  Q_D,D_D.OR.R[5]
U 1340. 0C01.003C.D980.0990.0084.7341   ;3447  SC_K[.9],RC[2]_D,D_Q
U 1341. 0D00.003C.0180.0800.0000.1342   ;3448  D_DAL.SC
U 1342. 0001.003C.0180.09E0.0000.1343   ;3449  RC[0C]_D    ; SAVE FOR TYPEOUT
U 1343. 0018.0038.D580.0998.0000.1344   ;3450  RC[3]_K[.6]    ; SET THE LOOP COUNT
U 1344. 0010.0038.0180.0908.0200.1345   ;3451  VA_ALU,ALU_RC[1]    ; LOAD MAP ADDRESS
U 1345. 0810.0038.0180.0910.0000.1346   ;3452  D_RC[2]    ; GET DATA FOR MAP
U 1346. 0000.003C.0180.A800.0000.10D8   ;3453  CACHE.P_D[LONG]    ; LOAD THE MAP
U 10D8. 0000.003D.0180.0800.0000.1163   ;3454 =0 ERLOOP
U 10D9. 0010.0038.0180.0958.0200.1347   ;3455 LOOP6: VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 1347. 0000.003C.0180.9000.0000.1348   ;3456  CACHE.INVALIDATE  ; MAKE IT A MISS
U 1348. 0010.0038.0180.0900.0200.1349   ;3457  VA_ALU,ALU_RC[0]  ; LOAD THE UNIBUS ADDRESS
U 1349. 0000.403C.0180.A800.0000.134A   ;3458  CACHE.P_D[WORD]    ; LOAD THE BUFFERED DATA PATH
U 134A. 0000.003C.0180.0A18.0200.134B   ;3459  VA_ALU,ALU_R[3]    ; LOAD ADDRESS OF DPR1
U 134B. 0000.003C.8D80.0800.0084.734C   ;3460  SC_K[.1F]
U 134C. 0818.0038.6580.0800.0000.134D   ;3461  D_K[.10]    ; WAIT FOR BDP TO FINISH
U 134D. 0819.0000.0580.0800.0010.134E   ;3462 BDPW3: D_D-K[.1],CLK.UBCC  ; LOADING
U 134E. 0000.013C.0180.0800.0000.10DA   ;3463  Z?
U 10DA. 0000.003C.0180.0800.0000.134D   ;3464 =0 J/BDPW3    ; WAIT SOME MORE
U 10DB. 0803.001C.0180.0800.0000.134F   ;3465  D_NOT.MASK
U 134F. 0000.003C.0180.A800.0000.1350   ;3466  CACHE.P_D[LONG]    ; START THE UBA
U 1350. 0000.003C.0180.0800.0000.1351   ;3467  NOP
U 1351. 0000.003C.0180.0800.0000.1352   ;3468  NOP
U 1352. 0000.003C.0180.0800.0000.1353   ;3469  NOP
U 1353. 0000.003C.0180.0800.0000.1354   ;3470  NOP
U 1354. 0000.003C.0180.0800.0000.1355   ;3471  NOP
U 1355. 0010.0038.0180.0958.0200.1356   ;3472  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 1356. 0000.003C.0180.C800.0000.1357   ;3473  D[LONG]_CACHE.P    ; START THE CPU READ
U 1357. 0000.003C.0180.C800.0000.1358   ;3474  D[LONG]_CACHE.P    ; SHOULD BE A HIT
U 1358. 0810.0038.75F0.2D70.0000.1359   ;3475  Q_ID[MAINT],D_RC[0E]  ; READ THE HIT/MISS BITS
U 1359. 001C.0034.01C0.0A38.0000.135A   ;3476  Q_Q.AND.R[7]    ; MASK
U 135A. 001D.2000.0180.0800.0010.135B   ;3477  ALU_Q-D,CLK.UBCC  ; CHECK FOR A HIT
U 135B. 0000.013C.0180.0800.0000.10DC   ;3478  Z?
U 10DC. 0001.203D.0180.09E8.0000.1166   ;3479 =0 ERROR2,RC[0D]_Q    ; NO HIT
                ;3480
U 10DD. 0810.0038.8580.0958.0000.135C   ;3481  D_RC[0B],K[.C]    ; GET THE CPU ADDRESS
U 135C. 001B.0000.85F8.0800.0082.135D   ;3482  SC_0-K[.C],Q_0
U 135D. 0D00.003C.0180.0800.0000.135E   ;3483  D_DAL.SC
U 135E. 0000.003C.85E0.0800.0084.735F   ;3484  Q_D,SC_K[.C]
U 135F. 0000.003C.03A8.0800.0000.1360   ;3485  Q_Q.LEFT,SI/MUL-
U 1360. 0C00.003C.05F8.0800.0084.B361   ;3486  D_Q,SC_SC-K[.1],Q_0
U 1361. 0D00.003C.0180.0800.0000.1362   ;3487  D_DAL.SC
U 1362. 0001.003C.0180.09D8.0000.1363   ;3488  RC[0B]_D    ; SAVE CPU ADDRESS (FLOATED A ZERO)
U 1363. 0810.0038.0180.0918.0000.1364   ;3489  D_RC[3]    ; GET THE LOOP COUNT
U 1364. 0019.0000.0580.0998.0010.1365   ;3490  RC[3]_D-K[.1],CLK.UBCC :
U 1365. 0000.013C.0180.0800.0000.10DE   ;3491  Z?
U 10DE. 0000.003C.0180.0800.0000.10D9   ;3492 =0 J/LOOP6
```

```
U 10DF, 0000.003C.0180.0800.0000.10E0  ;3493  NOP
          ;3494
          ;3495
          ;3496 ;////////////////////////////////////////////////////////////////////////
          ;3497 ;*
          ;3498 ; NOW PUT ALL ONE'S ON THE B LEG AND FLOAT A ZERO ACROSS THE A LEG.
          ;3499 ;-
          ;3500
U 10E0, 0000.003D.0180.0800.0000.117D  ;3501 =0 SUBTEST
U 10E1, 0818.0038.5580.0800.0000.1366  ;3502  D_K[.3F]
U 1366, 0000.003C.85F8.0800.0084.7367  ;3503  SC_K[.C],Q_0
U 1367, 0D00.003C.0180.0800.0000.1368  ;3504  D_DAL.SC
U 1368, 0001.003C.0180.09D8.0000.10E2  ;3505  RC[0B]_D    ; SAVE THE INIT CPU ADDRESS
U 10E2, 0818.0039.8D80.0800.0000.1168  ;3506 =0 D_K[.1F],CALL,J/DC0  ; GENERATE INIT MAP REGISTER DATA
U 10E3, 081C.2030.01E0.0A28.0000.1369  ;3507  Q_D,D_D.OR.R[5]
U 1369, 0001.003C.0180.0990.0000.136A  ;3508  RC[2]_D
U 136A, 0C00.003C.D9F8.0800.0084.736B  ;3509  D_Q,SC_K[.9],Q_0
U 136B, 0D00.003C.0180.0800.0000.136C  ;3510  D_DAL.SC
U 136C, 0001.003C.0180.09E0.0000.136D  ;3511  RC[0C]_D
U 136D, 0018.0038.D580.0998.0000.10E4  ;3512  RC[3]_K[.6]    ; SET THE LOOP COUNT
U 10E4, 0000.003D.0180.0800.0000.1163  ;3513 =0 ERLOOP
U 10E5, 0010.0038.0180.0908.0200.136E  ;3514 LOOP7: VA_ALU,ALU_RC[1]  ; LOAD THE MAP ADDRESS
U 136E, 0810.0038.0180.0910.0000.136F  ;3515  D_RC[2]     ; GET THE MAP DATA
U 136F, 0000.003C.0180.A800.0000.1370  ;3516  CACHE.P_D[LONG]    ; LOAD THE MAP
U 1370, 0010.0038.0180.0958.0200.1371  ;3517  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 1371, 0000.003C.0180.9000.0000.1372  ;3518  CACHE.INVALIDATE
U 1372, 0010.0038.0180.0900.0200.1373  ;3519  VA_ALU,ALU_RC[0]  ; LOAD THE UNIBUS ADDRESS
U 1373, 0000.403C.0180.A800.0000.1374  ;3520  CACHE.P_D[WORD]    ; LOAD THE BUFFERED DATA PATH
U 1374, 0000.003C.0180.0A18.0200.1375  ;3521  VA_ALU,ALU_R[3]    ; LOAD ADDRESS OF DPR1
U 1375, 0000.003C.8D80.0800.0084.7376  ;3522  SC_K[.1F]
U 1376, 0818.0038.6580.0800.0000.1377  ;3523  D_K[.10]    ; WAIT FOR BDP TO FINISH
U 1377, 0819.0000.0580.0800.0010.1378  ;3524 BDPW4: D_D-K[.1],CLK.UBCC  ; LOADING
U 1378, 0000.013C.0180.0800.0000.10E6  ;3525  Z?
U 10E6, 0000.003C.0180.0800.0000.1377  ;3526 =0 J/BDPW4     ; WAIT SOME MORE
U 10E7, 0803.001C.0180.0800.0000.1379  ;3527  D_NOT.MASK
U 1379, 0000.003C.0180.A800.0000.137A  ;3528  CACHE.P_D[LONG]    ; START THE UBA
U 137A, 0000.003C.0180.0800.0000.137B  ;3529  NOP
U 137B, 0000.003C.0180.0800.0000.137C  ;3530  NOP
U 137C, 0000.003C.0180.0800.0000.137D  ;3531  NOP
U 137D, 0000.003C.0180.0800.0000.137E  ;3532  NOP
U 137E, 0000.003C.0180.0800.0000.137F  ;3533  NOP
U 137F, 0010.0038.0180.0958.0200.1380  ;3534  VA_ALU,ALU_RC[0B]  ; LOAD THE CPU ADDRESS
U 1380, 0000.003C.0180.C800.0000.1381  ;3535  D[LONG]_CACHE.P    ; START THE CPU READ
U 1381, 0000.003C.0180.C800.0000.1382  ;3536  D[LONG]_CACHE.P    ; SHOULD BE A HIT
U 1382, 0810.0038.75F0.2D70.0000.1383  ;3537  Q_ID[MAINT],D_RC[0E]  ; READ THE HIT/MISS BITS
U 1383, 001C.0034.01C0.0A38.0000.1384  ;3538  Q_Q.AND.R[7]    ; MASK
U 1384, 001D.2000.0180.0800.0010.1385  ;3539  ALU_Q-D,CLK.UBCC  ; CHECK
U 1385, 0000.013C.0180.0800.0000.10E8  ;3540  Z?
U 10E8, 0001.203D.0180.09E8.0000.1166  ;3541 =0 ERROR2,RC[0D]_Q    ; CPU ADDRESS NOT A HIT
          ;3542
U 10E9, 0010.0038.01C0.0910.0000.1386  ;3543  Q_RC[2]     ; GET THE CONTENTS OF THE MAP
U 1386, 0000.003C.01A8.0800.0000.1387  ;3544  Q_Q.LEFT
U 1387, 0019.2014.01C0.0800.0000.10EA  ;3545  Q_Q+K[.8]
U 10EA, 0818.0039.8D80.0800.0000.1177  ;3546 =0 D_K[.1F],CALL,J/DCF
U 10EB, 081D.0034.0180.0A28.0000.1388  ;3547  D_D.AND.Q,LAB_R[5]
```

```
U 1388. 001C.2030.0180.0990.0000.1389  ;3548  RC[2]_ALU,ALU_LA[OR]D
U 1389. 0000.003C.D9F8.0800.0084.738A  ;3549  SC_K[.9].Q_0
U 138A. 0D00.003C.0180.0800.0000.138B  ;3550  D_DAL.SC
U 138B. 0001.003C.0180.09E0.0000.138C  ;3551  RC[0C]_D    ; SAVE FOR TYPEOUT
U 138C. 0810.0038.0180.0918.0000.138D  ;3552  D_RC[3]     ; GET THE LOOP COUNT
U 138D. 0019.0000.0580.0998.0010.138E  ;3553  RC[3]_D-K[.1],CLK.UBCC   ; CHECK
U 138E. 0000.013C.0180.0800.0000.10EC  ;3554  Z?
U 10EC. 0000.003C.0180.0800.0000.10E5  ;3555 =0 J/LOOP7
U 10ED. 0000.003C.0180.0800.0000.10F5  ;3556  J/END005
            ;3557
            ;3558 =0
            ;3559 END004:
U 10EE. 0000.003D.0180.0800.0000.124F  ;3560  CALL[DECFLG]
U 10EF. 0000.003C.0180.0800.0000.10F0  ;3561  NOP
U 10F0. 0818.0039.0580.0800.0000.124B  ;3562 =0 TYP_MSG_1_CRLF    ;TYPE THE NO MEMORY MESSAGE
U 10F1. 0000.003C.0180.0800.0000.10F5  ;3563  J/END005
            ;3564 =0
U 10F2. 0000.003D.0180.0800.0000.124F  ;3565 END003: CALL[DECFLG]
U 10F3. 0000.003C.0180.0800.0000.10F4  ;3566  NOP
U 10F4. 0818.0039.0980.0800.0000.124B  ;3567 =0 TYP_MSG_2_CRLF    ;TYPE THE NO UBA MESSAGE
U 10F5. 0000.003C.0180.0800.0000.10F6  ;3568 END005: NOP
            ;3569
            ;3570
```

```
;3571 .PAGE "TEST 208 SBI REQUESTS 4, 5, 6, OR 7"
;3572 ;*****************************************************************
;3573 ;++
;3574 ;
;3575 ; SBI REQUESTS 4, 5, 6, OR 7
;3576 ;
;3577 ; TEST DESCRIPTION
;3578 ;
;3579 ; THIS TEST CHECKS THE SBI INTERRUPTS AT LEVELS 4, 5, 6, OR 7
;3580 ; DEPENDING ON THE ASSIGNED LEVELS OF THE DEVICES AVAILABLE.
;3581 ;
;3582 ; THE FIRST PART OF THE TEST SEARCHES THE SBI FOR DEVICES. IF A DEVICE
;3583 ; IS FOUND, THE DEVICE CODE IS CHECKED TO DETERMINE WETHER IT'S A
;3584 ; UBA OR MBA. WHEN THIS DETERMINATION IS MADE, EXECUTION GOES TO
;3585 ; THE APPROPRIATE ROUTINE TO CAUSE THAT DEVICE TO INTERRUPT. ONCE THE
;3586 ; INTERRUPT IS ACTIVE, THE CPU INTERRUPT LOGIC IS TESTED INDEPENDENT
;3587 ; OF THE DEVICE TYPE.
;3588 ;
;3589 ; THE UBA ROUTINE SETS AND CLEARS THE UNIBUS POWER DOWN BIT IN
;3590 ; CONTROL REGISTER 1 WITH THE INTERRUPT ENABLE BIT SET. THIS SHOULD
;3591 ; CAUSE A UBA INTERRUPT. THE ROUTINE THEN READS THE UBA VECTOR REGISTERS
;3592 ; TO DETERMINE WHAT LEVEL THE INTERRUPT IS ACTIVE ON. THIS LEVEL IS
;3593 ; THEN CHECKED AGAINST THE IPL ACTIVE LEVEL IN THE CPU.
;3594 ;
;3595 ; THE MBA ROUTINE SETS THE MAINTENANCE MODE AND INTERRUPT ENABLE BITS
;3596 ; IN MBA REGISTER 1 AND THEN SETS AND CLEARS THE "SIMULATE ATTENTION"
;3597 ; BIT IN REGISTER 4. THIS SHOULD CAUSE AN INTERRUPT. THE ROUTINE
;3598 ; THEN READS THE IPL ACTIVE LEVEL AND ASSUMES THAT THIS IS THE
;3599 ; CORRECT INTERRUPT LEVEL FOR THIS DEVICE. IT IS CHECKED TO ENSURE
;3600 ; THAT IT IS BETWEEN 14 AND 17.
;3601 ;
;3602 ; SUBTEST 1 - UBA--CHECK IPL ACTIVE
;3603 ;      MBA--DO NOTHING
;3604 ;
;3605 ; SUBTEST 2 - CHECK THE INTERRUPT VECTOR FOR THE ACTIVE INTERRUPT LEVEL
;3606 ;
;3607 ; SUBTEST 3 - CHECK THE INTERRUPT VECTOR WHILE FLOATING A ONE ACROSS
;3608 ;      THE "PRIORITY" BITS (OF THE VECTOR REGISTER)
;3609 ;
;3610 ; SUBTEST 4 - CHECK THE INTERRUPT REQUEST BRANCH
;3611 ;
;3612 ; SUBTEST 5 - CHECK THE IB SERVICE BITS WITH THE INTERRUPT ACTIVE.
;3613 ;
;3614 ; SUBTEST 6 - ASSERT A RECEIVER INTERRUPT AND CHECK THE INTERRUPT REQUEST BRANCH
;3615 ;
;3616 ; LOGIC DESCRIPTION
;3617 ;
;3618 ; THIS TEST CHECKS MORE OF THE VECTOR ROM, THE "EXT INTR REQ", AND
;3619 ; THE VECTOR BITS ON THE ICL MODULE.
;3620 ;
;3621 ; ERROR DESCRIPTION
;3622 ;
;3623 ; SUBTEST 1 - DATA: EXPECTED IPL ACTIVE LEVEL
;3624 ;     RECEIVED IPL ACTIVE LEVEL
;3625 ;     NOT USED
```

```
            ;3626 ;     UBA/MBA FLAG
            ;3627 ;     BASE ADDRESS OF DEVICE
            ;3628 ;
            ;3629 ; SUBTEST 2,3 DATA: EXPECTED VECTOR BITS
            ;3630 ;     RECEIVED VECTOR BITS
            ;3631 ;     LOOP COUNT -- SUBTEST 3 ONLY
            ;3632 ;     UBA/MBA FLAG
            ;3633 ;     BASE ADDRESS OF DEVICE
            ;3634 ;
            ;3635 ; SUBTEST 4,6 DATA: EXPECTED BRANCH BITS <2:0>
            ;3636 ;     RECEIVED BRANCH BITS <2:0>
            ;3637 ;     NOT USED
            ;3638 ;     UBA/MBA FLAG
            ;3639 ;     BASE ADDRESS OF DEVICE
            ;3640 ;
            ;3641 ; SUBTEST 5 - DATA: EXPECTED CALL3 CODE
            ;3642 ;     RECEIVED CALL3 CODE
            ;3643 ;     NOT USED
            ;3644 ;     UBA/MBA FLAG
            ;3645 ;     BASE ADDRESS OF DEVICE
            ;3646 ;
            ;3647 ; NOTE: THE UBA/MBA FLAG IS = 0 FOR A UBA AND = 1 FOR A MBA
            ;3648 ;       THE CALL3 CODE REPRESENTS THE MICRO ADDRESS THAT WAS GENERATED
            ;3649 ;       FOR THE IB CALL3. I.E. 0=20, 1=24, 2=28, 3=2C, 4=30, 5=34,
            ;3650 ;       6=38, AND 7=3C.
            ;3651 ;--
            ;3652 ;*****************************************************************************
            ;3653
            ;3654 =0
U 10F6, 0000,003D,0180,0800,0000,105B  ;3655 T004: NEWTST
U 10F7, 0000,003C,0180,0800,0000,10F8  ;3656  NOP
U 10F8, 0000,003D,0180,0800,0000,104C  ;3657 =0 CALL,J/MAINTREGSETUP
U 10F9, 0F00,003C,0180,0800,0000,138F  ;3658  D_0
U 138F, 0000,003C,3580,3C00,0000,1390  ;3659  ID[VECTOR]_D   ; CLEAR THE VECTOR REGISTER
U 1390, 0000,003C,3D80,3C00,0000,10FA  ;3660  ID[PSL]_D    ; CLEAR THE PSL
U 10FA, 0000,003D,0180,0800,0000,1213  ;3661 =0 CALL,J/LOOKFORDEVICE  ; TRY AND FIND A UBA
            ;3662 CHECKR0A:
U 10FB, 001B,0010,1180,09F8,0000,1391  ;3663  RC[0F]_ALU,ALU_0+K[.4]+1
U 1391, 0000,003C,0180,0A00,0210,1392  ;3664  VA_ALU,ALU_R[0],CLK.UBCC ; HAS THERE A DEVICE?
U 1392, 0000,013C,0180,0800,0000,10FC  ;3665  Z?     ; BRANCH IF NO
U 10FC, 0000,003C,0180,0800,0000,1393  ;3666 =0 J/CHECKCODE1A    ; CHECK IF ITS A UBA
U 10FD, 0000,003C,0180,0800,0000,13ED  ;3667  J/END006    ; NO UBA AVAILABLE
            ;3668
            ;3669 CHECKCODE1A:
U 1393, 0800,003C,0180,0A00,0000,1394  ;3670  D_R[0]
U 1394, 0001,003C,0180,09D0,0000,1395  ;3671  RC[0A]_D    ; SAVE BASE ADDR OF DEVICE
U 1395, 0000,003C,0180,C800,0000,1396  ;3672  D[LONG]_CACHE.P    ; READ THE CONFIG REG
U 1396, 0200,003C,0180,0800,0000,1397  ;3673  D_D.RIGHT2    ; CHECK BITS 3 AND 4
U 1397, 0600,003C,0180,0800,0000,1398  ;3674  D_D.RIGHT    ; ...
U 1398, 0000,193C,0180,0800,0000,106C  ;3675  D1-0?     ; UBA = 01
U 106C, 0000,003C,0180,0800,0000,13AD  ;3676 =1100 J/MBASETUP    ; SETUP FOR MBA INTERRUPT
            ;3677
            ;3678
            ;3679 ;*
            ;3680 ; UBA SETUP ROUTINE. FORCES AN INTERRUPT FROM A UBA
```

```
            ;3681 ;-
            ;3682
U 106D, 0003.003C.0180.09D8.0000.106E  ;3683  RC[0B]_0   ; SET THE UBA FLAG
U 106E, 0800.003C.0180.0A00.0000.106F  ;3684  D_R[0]
U 106F, 0019.0014.1180.0800.0200.1399  ;3685  VA_ALU,ALU_D+K[.4]  ; LOAD ADDR OF REGISTER 1
U 1399, 0818.0038.0980.0800.0000.139A  ;3686  D_K[.2]
U 139A, 0000.003C.0180.A800.0000.139B  ;3687  CACHE.P_D[LONG]   ; SET THE POWER FAIL BIT
U 139B, 0818.0038.1180.0800.0000.139C  ;3688  D_K[.4]
U 139C, 0000.003C.0180.A800.0000.10FE  ;3689  CACHE.P_D[LONG]   ; CLR POWER FAIL AND SET IE
U 10FE, 0000.003D.0180.0800.0000.123F  ;3690 =0 CALL,J/WAITFORUNIBUS  ; WAIT FOR THE UNIBUS TO POWER UP
U 10FF, 0000.003C.1180.0800.0084.739D  ;3691  SC_K[.4]   ; FIND THE BR LEVEL
U 139D, 0800.003C.7980.0A00.0000.139E  ;3692  D_R[0],K[.30]   ; GET CONFIG REG ADDRESS
U 139E, 0819.0014.7980.0980.0000.139F  ;3693  D_D+K[.30],RC[0]_ALU  ; LOAD ADDR OF BR 4 VECTOR REG
            ;3694 TRYNEXTREGISTER:
U 139F, 0010.0038.0180.0900.0200.13A0  ;3695  VA_ALU,ALU_RC[0]
U 13A0, 0000.003C.01FB.C800.0000.13A1  ;3696  D[LONG]_CACHE.P,Q_0  ; READ THE REGISTER
U 13A1, 0000.003C.0128.0800.0000.13A2  ;3697  Q_Q.LEFT,SI/ASHL   ; GET BIT 31
U 13A2, 0001.203C.0180.0800.0010.13A3  ;3698  ALU_Q,CLK.UBCC   ; WAS IT SET?
U 13A3, 0000.013C.0180.0800.0000.110A  ;3699  Z?
U 110A, 0000.003C.0180.0800.0000.1112  ;3700 =0 J/FOUNDBRLEVEL   ; FOUND THE INTERRUPT LEVEL
U 110B, 0000.003C.0580.0800.0084.93A4  ;3701  SC_SC+K[.1]   ; INCREMENT THE BR LEVEL COUNTER
U 13A4, 0803.001C.0180.0800.0000.13A5  ;3702  D_NOT.MASK
U 13A5, 0001.803C.0180.0800.0010.13A6  ;3703  ALU_D,CLK.UBCC,BYTE  ; OVERFLOW YET?
U 13A6, 0810.0138.0180.0900.0000.1110  ;3704  Z?,D_RC[0]
U 1110, 0019.0014.1180.0980.0000.139F  ;3705 =0 J/TRYNEXTREGISTER,RC[0]_D+K[.4] ; CHECK THE NEXT BR LEVEL
U 1111, 0000.003D.0180.0800.0000.1164  ;3706  ERROR1    ; NO INTERRUPT FROM UBA
            ;3707
            ;3708
            ;3709 ;////////////////////////////////////////////////////////////////////////////
            ;3710 ;+
            ;3711 ; CHECK THAT THE UBA INTERRUPT LEVEL MATCHES THE IPL ACTIVE LEVEL
            ;3712 ;-
            ;3713
            ;3714 =0
            ;3715 FOUNDBRLEVEL:
U 1112, 0000.003D.0180.0800.0000.117D  ;3716  SUBTEST
U 1113, 0818.0038.1DB0.0800.0000.13A7  ;3717  D_SC    ; MOVE THE BR LEVEL TO THE IPL
U 13A7, 0819.0030.65F8.0800.0084.73A8  ;3718  D_D.OR.K[.10],SC_K[.10],Q_0 ; ACTIVE BIT POSITIONS
U 13A8, 0D00.003C.0180.0800.0000.13A9  ;3719  D_DAL.SC
U 13A9, 0001.003C.0180.09F0.0000.1114  ;3720  RC[0E]_D   ; SAVE AS EXPECTED IPL ACTIVE
U 1114, 0000.003D.0180.0800.0000.1163  ;3721 =0 ERLOOP
U 1115, 0000.003C.0180.0800.4000.13AA  ;3722  INTRPT.STROBE   ; LATCH THE INTERRUPT
U 13AA, 0810.0038.39F0.2D70.0000.13AB  ;3723  Q_ID[SIR],D_RC[0E]   ; READ THE IPL ACTIVE
U 13AB, 001D.2000.0180.0800.0010.13AC  ;3724  ALU_Q-D,CLK.UBCC   ; ARE THEY THE SAME?
U 13AC, 0000.013C.0180.0800.0000.1116  ;3725  Z?
U 1116, 0001.203D.0180.09E8.0000.1166  ;3726 =0 ERROR2,RC[0D]_Q   ; IPL ACTIVE NOT SAME AS BR LEVEL
U 1117, 0000.003C.0180.0800.0000.111E  ;3727  J/T001S2   ; GO TO SUBTEST 2
            ;3728
            ;3729
            ;3730
            ;3731 ;+
            ;3732 ; SETUP THE MBA TO INTERRUPT
            ;3733 ;-
            ;3734
            ;3735 MBASETUP:
```

```
U 13AD, 0018.0038.0580.09D8.0000.13AE  ;3736  RC[0B]_K[.1]
U 13AE, 0800.003C.0180.0A00.0000.13AF  ;3737  D_R[0]
U 13AF, 0019.0014.1180.0800.0200.13B0  ;3738  VA_ALU,ALU_D+K[.4]  ; LOAD ADDR OF REG 1
U 13B0, 0818.0038.0580.0800.0000.1118  ;3739  D_K[.1]
U 1118, 0000.003D.0180.A800.0000.123F  ;3740 =0 CACHE.P_D[LONG],CALL[WAITFORUNIBUS] ; INIT THE MBA
U 1119, 0818.0038.F980.0803.0000.13B1  ;3741  VA_VA+4,D_K[.7E]  ; LOAD ADDRESS OF REG 2
U 13B1, 0819.0034.6180.0800.0000.13B2  ;3742  D_D.AND.K[.F]
U 13B2, 0000.003C.65F8.0800.0084.73B3  ;3743  SC_K[.10],Q_0
U 13B3, 0D00.003C.0180.0800.0000.13B4  ;3744  D_DAL.SC   ; SETUP TO CLEAR BITS 17, 18 AND 19
U 13B4, 0000.003C.0180.A800.0000.13B5  ;3745  CACHE.P_D[LONG]   ; ...
U 13B5, 0800.003C.0180.0A00.0000.13B6  ;3746  D_R[0]
U 13B6, 0019.0014.1180.0800.0200.13B7  ;3747  VA_ALU,ALU_D+K[.4]  ; LOAD ADDR OF REG 1
U 13B7, 0818.0038.8580.0800.0000.13B8  ;3748  D_K[.C]
U 13B8, 0000.003C.0180.A800.0000.13B9  ;3749  CACHE.P_D[LONG]   ; SET MMM AND IE BITS
U 13B9, 0800.003C.2180.0A00.0000.13BA  ;3750  D_R[0],K[.14]
U 13BA, 0019.0014.2180.0800.0200.13BB  ;3751  VA_ALU,ALU_D+K[.14]  ; LOAD ADDR OF REG 5
U 13BB, 0818.0038.0580.0800.0000.13BC  ;3752  D_K[.1]
U 13BC, 0B00.003C.0180.0800.0000.13BD  ;3753  D_D.SWAP
U 13BD, 0000.003C.0180.A800.0000.13BE  ;3754  CACHE.P_D[LONG]   ; SET SIMULATE ATTENTION BIT
U 13BE, 0818.0038.6580.0800.0000.13BF  ;3755  D_K[.10]   ; WAIT FOR THE INTERRUPT
            ;3756 WAITFORMBA:
U 13BF, 0819.0000.0580.0800.0010.13C0  ;3757  D_D-K[.1],CLK.UBCC
U 13C0, 0000.013C.0180.0800.0000.111A  ;3758  Z?
U 111A, 0000.003C.0180.0800.0000.13BF  ;3759 =0 J/WAITFORMBA
U 111B, 0F00.003C.0180.0800.4000.13C1  ;3760  INTRPT.STROBE,D_0  ; LATCH THE INTERRUPT
U 13C1, 0000.003C.3D80.3C00.0000.111C  ;3761  ID[PSL]_D
            ;3762
            ;3763 ;*
            ;3764 ; READ THE IPL ACTIVE AND SAVE AS THE INTERRUPT LEVEL
            ;3765 ;-
            ;3766
U 111C, 0000.003D.0180.0800.0000.117D  ;3767 =0 SUBTEST
U 111D, 0000.003C.39F0.2C00.0000.13C2  ;3768  Q_ID[SIR]   ; READ THE IPL ACTIVE
U 13C2, 0001.203C.0180.09F0.0000.111E  ;3769  RC[0E]_Q
            ;3770
            ;3771
            ;3772 ;////////////////////////////////////////////////////////////////////////
            ;3773 ;*
            ;3774 ; CHECK THAT THE VECTOR IS CORRECT FOR THIS INTERRUPT LEVEL
            ;3775 ;-
            ;3776
            ;3777 =0
U 111E, 0000.003D.0180.0800.0000.117D  ;3778 T001S2: SUBTEST
U 111F, 0810.0038.65F8.0970.0000.13C3  ;3779  D_RC[0E],K[.10],Q_0  ; GET THE IPL ACTIVE LEVEL
U 13C3, 001B.0000.6580.0800.0082.13C4  ;3780  SC_0-K[.10]   ; SET THE SHIFT COUNT
U 13C4, 0D18.0038.41C0.0800.0000.1120  ;3781  D_DAL.SC,Q_K[.80]
U 1120, 0000.193D.01A8.0800.0000.1078  ;3782 =0 D3-0?,Q_Q.LEFT,CALL,J/GENVECTOR ; GENERATE THE VECTOR FOR THIS IPL ACTIVE
U 1121, 0001.203C.0180.09F0.0000.1122  ;3783  RC[0E]_Q   ; SAVE THE EXPECTED VECTOR
U 1122, 0000.003D.0180.0800.0000.1163  ;3784 =0 ERLOOP
U 1123, 0810.0038.35F0.2D70.0000.13C5  ;3785  Q_ID[VECTOR],D_RC[0E]  ; READ THE VECTOR
U 13C5, 001D.2000.0180.0800.0010.13C6  ;3786  ALU_Q-D,CLK.UBCC  ; IS IT CORRECT?
U 13C6, 0000.013C.0180.0800.0000.1124  ;3787  Z?
U 1124, 0001.203D.0180.09E8.0000.1166  ;3788 =0 ERROR2,RC[0D]_Q   ; VECTOR INCORECT FOR THIS IPL LEVEL
U 1125, 0000.003C.0180.0800.0000.1126  ;3789  NOP
            ;3790
```

```
            ;3791 ;////////////////////////////////////////////////////////////////////////
            ;3792 ;*
            ;3793 ; CHECK THE VECTOR WITH A FLOATING ONE IN THE PRIORITY BITS
            ;3794 ;-
            ;3795
U 1126. 0000.003D.0180.0800.0000.117D  ;3796 =0 SUBTEST
U 1127. 0810.0038.0180.0970.0000.13C7  ;3797  D_RC[0E]    ; GET THE BASE VECTOR
U 13C7. 0001.003C.6580.0980.0084.73C8  ;3798  RC[0]_D,SC_K[.10]  ; SAVE
U 13C8. 0818.0038.09F8.0800.0000.13C9  ;3799  D_K[.2],Q_0    ; INIT PATTERN TO LOAD VECTOR REG
U 13C9. 0D00.003C.0180.0800.0000.13CA  ;3800  D_DAL.SC
U 13CA. 0001.003C.0180.0988.0000.13CB  ;3801  RC[1]_D     ; SAVE
U 13CB. 0018.0038.1180.09E0.0000.13CC  ;3802  RC[0C]_K[.4]    ; SET THE LOOP COUNT
U 13CC. 0018.0038.11C0.0990.0000.13CD  ;3803  RC[2]_K[.4],Q_ALU  ; SET INIT DATA TO OFFSET BASE VECTOR
U 13CD. 0810.0038.0180.0900.0000.13CE  ;3804  D_RC[0]     ; GET BASE VECTOR
U 13CE. 001D.0014.0180.09F0.0000.13CF  ;3805  RC[0E]_D+Q    ; GENERATE EXPECTED VECTOR
U 13CF. 0000.003C.0580.0800.0084.7128  ;3806  SC_K[.1]    ; SET SHIFT CNT FOR PRIORITY BITS
U 1128. 0000.003D.0180.0800.0000.1163  ;3807 =0 ERLOOP
U 1129. 0810.0038.0180.0908.0000.13D0  ;3808 LOOPA: D_RC[1]     ; GET DATA TO LOAD VECTOR REG
U 13D0. 0000.003C.3580.3C00.0000.13D1  ;3809  ID[VECTOR]_D    ; SETUP THE PRIORITY BITS
U 13D1. 0810.0038.35F0.2D70.0000.13D2  ;3810  Q_ID[VECTOR],D_RC[0E]  ; READ THE RESULTANT VECTOR
U 13D2. 0019.2034.C1C0.0800.0000.13D3  ;3811  Q_Q.AND.K[.FFFF]  ; MASK
U 13D3. 001D.2000.0180.0800.0010.13D4  ;3812  ALU_Q-D,CLK.UBCC  ; IS IT CORRECT?
U 13D4. 0000.013C.0180.0800.0000.112A  ;3813  Z?
U 112A. 0001.203D.0180.09E8.0000.1166  ;3814 =0 ERROR2,RC[0D]_Q    ; VECTOR INCORRECT WITH PRIORITY
            ;3815       ; BITS OR'D IN
            ;3816
            ;3817 ;*
            ;3818 ; THE FOLLOWING IS LOOP CONTROL FOR THIS SUBTEST
            ;3819 ;-
            ;3820
U 112B. 0810.0038.0180.0910.0000.13D5  ;3821  D_RC[2]     ; GET THE CURRENT VECTOR OFFSET
U 13D5. 0510.0038.01C0.0900.0000.13D6  ;3822  D_D.LEFT,Q_RC[0]
U 13D6. 0001.003C.0180.0990.0000.13D7  ;3823  RC[2]_D     ; SAVE THE NEXT OFFSET
U 13D7. 001D.0014.0180.09F0.0000.13D8  ;3824  RC[0E]_D+Q    ; SET THE EXPECTED VECTOR
U 13D8. 0810.0038.01F8.0908.0000.13D9  ;3825  D_RC[1],Q_0    ; GET DATA TO LOAD VECTOR REG
U 13D9. 0D00.003C.0180.0800.0000.13DA  ;3826  D_DAL.SC    ; GEN NEXT DATA PATTERN
U 13DA. 0001.003C.0180.0988.0000.13DB  ;3827  RC[1]_D     ; SAVE
U 13DB. 0818.0038.1D80.0800.0000.13DC  ;3828  D_SC     ; GET THE SHIFT COUNT OF THE PATTERN
U 13DC. 0500.003C.0180.0800.0000.13DD  ;3829  D_D.LEFT
U 13DD. 0001.003C.0180.0800.0082.13DE  ;3830  SC_D     ; SET THE NEXT SHIFT COUNT
U 13DE. 0810.0038.0180.0960.0000.13DF  ;3831  D_RC[0C]    ; GET THE LOOP COUNT
U 13DF. 0019.0000.0580.09E0.0010.13E0  ;3832  RC[0C]_D-K[.1],CLK.UBCC  ; DONE YET?
U 13E0. 0000.013C.0180.0800.0000.112C  ;3833  Z?
U 112C. 0000.003C.0180.0800.0000.1129  ;3834 =0 J/LOOPA     ; CONTINUE LOOP
U 112D. 0000.003C.0180.0800.0000.112E  ;3835  NOP
            ;3836
            ;3837
            ;3838 ;////////////////////////////////////////////////////////////////////////
            ;3839 ;*
            ;3840 ; CHECK THE INTERRUPT REQUEST BRANCH -- BRANCH ENABLE E
            ;3841 ;-
            ;3842
U 112E. 0000.003D.0180.0800.0000.117D  ;3843 =0 SUBTEST
U 112F. 0003.003C.0180.09F0.0000.1130  ;3844  RC[0E]_0    ; SET THE EXPECTED BRANCH BITS
U 1130. 0000.003D.0180.0800.0000.1163  ;3845 =0 ERLOOP
```

```
U 1131, 0000,0E3C,0180,0800,0000,1005  ;3846  AC.LOW?
U 1005, 0000,003C,0180,0800,0000,1133  ;3847 =101 J/BRANCHOK   ; BRANCH IS OK
U 1007, 0000,003C,0180,0800,0000,1132  ;3848  NOP
U 1132, 0018,0039,0980,09E8,0000,1166  ;3849 =0 ERROR2,RC[0D]_K[.2]  ; BRANCH BIT 1 ASCERTED
          ;3850 BRANCHOK:
U 1133, 0000,003C,0180,0800,0000,1134  ;3851  NOP
          ;3852
          ;3853
          ;3854 ;////////////////////////////////////////////////////////////////////////
          ;3855 ;+
          ;3856 ; CHECK THE INSTRUCTION BUFFER SERVICE BITS
          ;3857 ;-
          ;3858
U 1134, 0000,003D,0180,0800,0000,117D  ;3859 =0 SUBTEST
U 1135, 0018,0038,5D80,09F0,0000,1136  ;3860  RC[0E]_K[.7]   ; SET THE EXPECTED CALL3 CODE
U 1136, 0000,003D,0180,0800,0000,1163  ;3861 =0 ERLOOP
U 1137, 2000,003C,0180,0800,0000,1138  ;3862  IBC/FLUSH   ; ENSURE THE IB IS CLEAR
U 1138, 0000,003D,0180,0800,0000,1182  ;3863 =0 CALL,J/DISPATIRD  ; GO EXECUTE THE CALL3
U 1139, 0810,0038,0180,0970,0000,13E1  ;3864  D_RC[0E]
U 13E1, 001D,2000,0180,0800,0010,13E2  ;3865  ALU_Q-D,CLK.UBCC  ; IS THE CALL3 CODE CORRECT?
U 13E2, 0000,013C,0180,0800,0000,113A  ;3866  Z?
U 113A, 0001,203D,0180,09E8,0000,1166  ;3867 =0 ERROR2,RC[0D]_Q   ; UPC INCORRECT ON CALL3
U 113B, 0000,003C,0180,0800,0000,113C  ;3868  NOP
          ;3869
          ;3870 ;////////////////////////////////////////////////////////////////////////
          ;3871 ;+
          ;3872 ; CHECK THE "EXT INTR REQ" SIGNAL WITH A RECEIVER INTERRUPT ACTIVE
          ;3873 ;-
          ;3874
U 113C, 0000,003D,0180,0800,0000,117D  ;3875 =0 SUBTEST
U 113D, 0003,003C,0180,09F0,0000,113E  ;3876  RC[0E]_0   ; SET THE EXPECTED BRANCH BITS
U 113E, 001B,0039,4180,0AD0,0000,1187  ;3877 =0 SETRXCS[.80]   ; SET THE READY BIT IN THE RXCS
U 113F, 081B,0038,3180,0800,0000,13E3  ;3878  D_K[.40]
U 13E3, 0000,003C,1180,3C00,0000,13E4  ;3879  ID[RXCS]_D   ; SET THE INTERRUPT ENABLE
U 13E4, 0000,003C,0180,0800,0000,13E5  ;3880  NOP
U 13E5, 0000,003C,0180,0800,4000,1140  ;3881  INTRPT.STROBE   ; LATCH THE INTERRUPT
U 1140, 0000,003D,0180,0800,0000,1163  ;3882 =0 ERLOOP
U 1141, 0000,0E3C,0180,0800,0000,1015  ;3883  AC.LOW?
U 1015, 0000,003C,0180,0800,0000,1143  ;3884 =101 J/BRANCHOK2   ; BRANCH IS OK
U 1017, 0000,003C,0180,0800,0000,1142  ;3885  NOP
U 1142, 001B,0039,0980,09E8,0000,1166  ;3886 =0 ERROR2,RC[0D]_K[.2]  ; BRANCH BIT ASCERTED
          ;3887 BRANCHOK2:
U 1143, 0F00,003C,0180,0800,0000,13E6  ;3888  D_0
U 13E6, 0000,003C,1180,3C00,0000,1144  ;3889  ID[RXCS]_D   ; CLEAR THE INTERRUPT ENABLE
U 1144, 0018,0039,1980,0AD0,8000,1187  ;3890 =0 INTRPT.ACK,SETRXCS[ZERO] ; CLEAR THE READY BIT & THE INTERRUPT
U 1145, 0000,003C,0180,0800,4000,1146  ;3891  INTRPT.STROBE   ;
          ;3892
          ;3893
          ;3894 ;+
          ;3895 ; SEE IF ALL THE TR LEVELS HAVE BEEN TRIED FOR A DEVICE
          ;3896 ;-
          ;3897
U 1146, 0000,003D,0180,0800,0000,1197  ;3898 =0 CALL,J/UNJAM   ; CLEAR THE INTERRUPTS
U 1147, 0F00,003C,0180,0800,0000,13E7  ;3899  D_0
U 13E7, 0000,003C,3580,3C00,0000,13E8  ;3900  ID[VECTOR]_D   ; CLEAR THE VECTOR REGISTER
```

```
U 13E8. 0000.003C.7580.3C00.0000.13E9  ;3901  ID[MAINT]_D   ; TURN ON THE SBI
U 13E9. 0000.003C.3D80.3C00.0000.13EA  ;3902  ID[PSL]_D    ; CLEAR THE IPL LEVEL
U 13EA. 0000.003C.0180.0800.0084.73EB  ;3903  SC_K[.8]
U 13EB. 0818.0038.75F8.0800.0000.13EC  ;3904  D_K[.20],Q_0
U 13EC. 0D00.003C.0180.0800.0000.1148  ;3905  D_DAL.SC
U 1148. 0000.003D.89E0.0800.0084.708B  ;3906 =0 Q_D,SC_K[.D],CALL,J/FOUNDMEMORY
U 1149. 0000.003C.0180.0800.0000.10FB  ;3907  J/CHECKR0A
            ;3908
            ;3909
            ;3910 ;*
            ;3911 ; THIS SUBROUTINE IS USED TO GENERATE THE EXPECTED INTERRUPT VECTOR
            ;3912 ;-
            ;3913
            ;3914 =1000
            ;3915 GENVECTOR:
U 1078. 0000.003C.0180.0800.0000.1079  ;3916  NOP
U 1079. 0000.003C.0180.0800.0000.107A  ;3917  NOP
U 107A. 0000.003C.0180.0800.0000.107B  ;3918  NOP
U 107B. 0000.003C.0180.0800.0000.107C  ;3919  NOP
U 107C. 0000.003E.0180.0800.0000.0001  ;3920  RETURN1     ; Q = 100
U 107D. 0019.2032.31C0.0800.0000.0001  ;3921  Q_Q.OR.K[.40],RETURN1
U 107E. 0019.2032.41C0.0800.0000.0001  ;3922  Q_Q.OR.K[.80],RETURN1
U 107F. 0019.2032.D1C0.0800.0000.0001  ;3923  Q_Q.OR.K[.C0],RETURN1   ;
            ;3924
U 13ED. 0000.003C.0180.0800.0000.114A  ;3925 END006: NOP
            ;3926
```

```
            ;3927  .PAGE "TEST 209 RESERVED"
            ;3928 =0
U 114A, 0000,003D,0180,0800,0000,105B  ;3929 T209: NEWTST
U 114B, 0000,003C,0180,0800,0000,114C  ;3930  NOP
            ;3931
```

```
            ;3932  .PAGE "TEST 20A RESERVED"
            ;3933 =0
U 114C, 0000,003D,0180,0800,0000,105B  ;3934 T20A: NEWTST
U 114D, 0000,003C,0180,0800,0000,114E  ;3935  NOP
            ;3936
```

```
            ;3937  .PAGE "DUMMY NEWTST"
            ;3938 =0
U 114E. 0000,003D,0180,0800,0000,105B  ;3939 DUM: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2069
ESKAR55.MCR                        MICRO2  1P(00)    26-JUL-85  17:09:04       Page  10
;                                Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1911: 1880 1882= 1901= 1881 3847= 1904 3848=
U 1008   2517= 2518= 2519= 2520= 2521= 2522= 2523= 2524=
U 1010   2427= 2428= 2429= 1905 1906 3884= 1907 3885=
U 1018   2902= 2903= 2904= 2905= 2906= 2907= 2908= 2909=
U 1020   2434= 2435= 2436= 1908 1902= 1903= 1909 2942=
U 1028   2440= 2441= 2442= 1910 1919= 1920= 1917 2943=
U 1030   2446= 2449= 2451= 2452= 2797= 2798= 2799= 2801=
U 1038   2846= 2847= 2848= 2850= 3228= 3229= 1974= 1976=
U 1040   2007= 2008= 2032= 2033= 2269= 2271= 2274= 2276=
U 1048   2279= 2281= 2307= 2308= 2336= 2337= 2014: 1918
U 1050   2338= 2339= 2348= 2349= 2351= 2352= 2353= 2354=
U 1058   2494= 2495= 2039: 1963 2496= 2497= 2043: 1964
U 1060   2505= 2506= 2507= 2509= 2512= 2513= 2554= 2555=
U 1068   2556= 2557= 2564= 2565= 3676= 3683= 3684= 3685=
U 1070   2571= 2573= 2576= 2577= 2597= 2598= 2600= 2601=
U 1078   3916= 3917= 3918= 3919= 3920= 3921= 3922= 3923=
U 1080   2693= 2694= 2697= 2698= 2709= 2710= 2713= 2714=
U 1088   2751= 2752= 2879= 2881= 2890= 2891= 2924= 2925=
U 1090   2998= 2999= 3000= 3001= 3013= 3014= 3039= 3040=
U 1098   3049= 3050= 3054= 3055= 3098= 3099= 3100= 3101=
U 10A0   3117= 3118= 3143= 3144= 3153= 3154= 3158= 3159=
U 10A8   3213= 3214= 3215= 3216= 3217= 3218= 3220= 3221=
U 10B0   3243= 3244= 3247= 3248= 3270= 3271= 3277= 3278=
U 10B8   3290= 3291= 3307= 3309= 3310= 3311= 3322= 3323=
U 10C0   3330= 3331= 3342= 3343= 3351= 3352= 3367= 3369=
U 10C8   3375= 3376= 3383= 3384= 3392= 3393= 3404= 3405=
U 10D0   3419= 3421= 3431= 3432= 3440= 3441= 3445= 3446=
U 10D8   3454= 3455= 3464= 3465= 3479= 3481= 3492= 3493=
U 10E0   3501= 3502= 3506= 3507= 3513= 3514= 3526= 3527=
U 10E8   3541= 3543= 3546= 3547= 3555= 3556= 3560= 3561=
U 10F0   3562= 3563= 3565= 3566= 3567= 3568= 3655= 3656=
U 10F8   3657= 3658= 3661= 3663= 3666= 3667= 3690= 3691=
U 1100   1866: 1867: 1868: 1869: 1870: 1871: 1872: 1873:
U 1108   1874: 1965 3700= 3701= 1875: 1876: 1877: 1878:
U 1110   3705= 3706= 3716= 3717= 3721= 3722= 3726= 3727=
U 1118   3740= 3741= 3759= 3760= 3767= 3768= 3778= 3779=
U 1120   3782= 3783= 3784= 3785= 3788= 3789= 3796= 3797=
U 1128   3807= 3808= 3814= 3821= 3834= 3835= 3843= 3844=
U 1130   3845= 3846= 3849= 3851= 3859= 3860= 3861= 3862=
U 1138   3863= 3864= 3867= 3868= 3875= 3876= 3877= 3878=
U 1140   3882= 3883= 3886= 3888= 3890= 3891= 3898= 3899=
U 1148   3906= 3907= 3929= 3930= 3934= 3935= 3939= 1966
U 1150   1967 1968 1969 1970 1971 2134: 1972 1973
U 1158   1977 1978 1979 1980 1981 1982 1983 1984
U 1160   1985 1986 1987 1989 2012 2013 2037 2038
U 1168   2081 2082 2083 2084 2085 2086 2087 2088
U 1170   2089 2090 2091 2092 2093 2094 2095 2096
U 1178   2098 2099 2100 2101 2103 2122 2123 2131
U 1180   2132 2133 2139 2167 2168 2169 2171 2172
U 1188   2173 2174 2176 2177 2178 2179 2180 2204
U 1190   2232 2233 2234 2254 2255 2256 2257 2264
U 1198   2265 2266 2267 2268 2272 2273 2277 2278
```

ZZ-ESKAR-V22 Line Number Index
ESKAR55.MCR MICRO2 1P(00) 26-JUL-85 17:09:04 SEQ 2070 Page 10

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2282 2283 2309 2310 2311 2312 2340 2341
U 11A8   2345 2347 2350 2355 2356 2357 2358 2359
U 11B0   2360 2361 2362 2363 2364 2365 2366 2367
U 11B8   2368 2369 2370 2371 2372 2373 2374 2375
U 11C0   2376 2377 2378 2379 2380 2381 2425 2426
U 11C8   2453 2498 2499 2500 2501 2510 2511 2515
U 11D0   2516 2529 2530 2532 2533 2534 2536 2541
U 11D8   2542 2543 2545 2550 2552 2553 2561 2562
U 11E0   2566 2567 2568 2569 2570 2578 2579 2580
U 11E8   2582 2583 2584 2585 2586 2587 2593 2595
U 11F0   2596 2599 2624 2625 2626 2627 2657 2658
U 11F8   2660 2661 2662 2689 2691 2692 2696 2700
U 1200   2701 2702 2703 2705 2706 2707 2708 2711
U 1208   2712 2740 2741 2742 2743 2744 2745 2746
U 1210   2747 2749 2750 2859 2860 2861 2862 2863
U 1218   2864 2865 2866 2867 2868 2869 2870 2871
U 1220   2141: 2872 2874 2875 2142: 2876 2877 2878
U 1228   2143: 2882 2883 2884 2144: 2885 2886 2887
U 1230   2145: 2888 2889 2893 2146: 2894 2895 2896
U 1238   2147: 2898 2899 2900 2148: 2901 2911 2920
U 1240   2921 2922 2923 2934 2936 2937 2938 2939
U 1248   2940 2941 2944 2946 2947 2948 2950 2955
U 1250   2956 2957 3002 3003 3004 3005 3006 3007
U 1258   3008 3009 3010 3011 3015 3016 3017 3018
U 1260   3019 3020 3021 3022 3023 3024 3025 3026
U 1268   3027 3028 3029 3030 3031 3032 3033 3034
U 1270   3035 3036 3037 3038 3041 3042 3043 3044
U 1278   3045 3046 3047 3048 3052 3053 3102 3103
U 1280   3104 3105 3106 3107 3108 3109 3110 3111
U 1288   3112 3113 3114 3115 3119 3120 3121 3122
U 1290   3123 3124 3125 3126 3127 3128 3129 3130
U 1298   3131 3132 3133 3134 3135 3136 3137 3138
U 12A0   3139 3140 3141 3142 3145 3146 3147 3148
U 12A8   3149 3150 2135: 3151 3152 3156 3157 3219
U 12B0   3224 3225 3226 3227 3230 3231 3232 3233
U 12B8   3234 3235 3236 3237 3240 3241 3242 3249
U 12C0   3250 3251 3252 3253 3254 3255 3256 3257
U 12C8   3258 3259 3260 3261 3262 3263 3272 3273
U 12D0   3274 3275 3276 3279 3280 3282 3283 3284
U 12D8   3285 3286 3287 3288 3289 3292 3293 3294
U 12E0   3295 3296 3297 3298 3299 3303 3304 3305
U 12E8   3306 3312 3313 3314 3315 3316 3317 3318
U 12F0   3319 3320 3321 3332 3333 3334 3335 3336
U 12F8   3337 3338 3339 3340 3341 3344 3345 3346
U 1300   3347 3348 3349 3350 3353 3354 3355 3356
U 1308   3357 3358 3359 3360 3362 3363 3364 3365
U 1310   3366 3370 3371 3372 3373 3374 3385 3386
U 1318   3387 3388 3389 3390 3391 3394 3395 3396
U 1320   3397 3398 3399 3400 3401 3402 3403 3406
U 1328   3407 3408 3409 3410 3411 3412 3413 3414
U 1330   3415 3416 3417 3418 3422 3423 3424 3425
U 1338   3426 3427 3428 3429 3430 3442 3443 3444
U 1340   3447 3448 3449 3450 3451 3452 3453 3456
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3457 3458 3459 3460 3461 3462 3463 3466
U 1350  3467 3468 3469 3470 3471 3472 3473 3474
U 1358  3475 3476 3477 3478 3482 3483 3484 3485
U 1360  3486 3487 3488 3489 3490 3491 3503 3504
U 1368  3505 3508 3509 3510 3511 3512 3515 3516
U 1370  3517 3518 3519 3520 3521 3522 3523 3524
U 1378  3525 3528 3529 3530 3531 3532 3533 3534
U 1380  3535 3536 3537 3538 3539 3540 3544 3545
U 1388  3548 3549 3550 3551 3552 3553 3554 3659
U 1390  3660 3664 3665 3670 3671 3672 3673 3674
U 1398  3675 3686 3687 3688 3689 3692 3693 3695
U 13A0  3696 3697 3698 3699 3702 3703 3704 3718
U 13A8  3719 3720 3723 3724 3725 3736 3737 3738
U 13B0  3739 3742 3743 3744 3745 3746 3747 3748
U 13B8  3749 3750 3751 3752 3753 3754 3755 3757
U 13C0  3758 3761 3769 3780 3781 3786 3787 3798
U 13C8  3799 3800 3801 3802 3803 3804 3805 3806
U 13D0  3809 3810 3811 3812 3813 3822 3823 3824
U 13D8  3825 3826 3827 3828 3829 3830 3031 3832
U 13E0  3833 3865 3866 3879 3880 3881 3889 3900
U 13E8  3901 3902 3903 3904 3905 3925
U 13F0  2049: 2050: 2051: 2052: 2053: 2054: 2055: 2056:
U 13F8  2057: 2058: 2059: 2060: 2061: 2062: 2063: 2064:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR55	1022
Used	1022
Remaining	

Total microwords used in memory U: 1022
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR55.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1842 MICROTRAP HANLDE AND LSI ROUTINES
; 1932 Micro Monitor Interface Routines
; 1933   Newtest Routine
; 1992   ERROR1 WITH PARAMETER
; 2017   ERROR 2 WITH PARAMETER
; 2066   Set Trap Mask
; 2095   Disable Cache Routine
; 2117   Initialize the SBI
; 2145   ASCII MESSAGES
; 2204   RESET SUBTEST NUMBER
; 2385   CONFIGURE MS780-E/H
; 2459   Find MS780-E Memory
; 2606   Generate IO Address
; 2633   Set Starting Address
; 2667   ENABLE NEXT MS780-E
; 2719   Select Controller
; 2758   SELECT LOWER CONTROLLER
; 2807   SELECT UPPER CONTROLLER
; 2961 TEST 20B SBI INVALIDATE
; 309C TEST 20C SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR0
; 3159 TEST 20D SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR1
; 3233 TEST 20E NO SBI INVALIDATE IN GR0 ON I/O WRITE CYCLE
; 3295 TEST 20F NO SBI INVALIDATE IN GR1 ON I/O WRITE CYCLE
; 3364 TEST 210 INTERRUPT SUMMARY READ TEST
; 3501 TEST 211 RESERVED
; 3506 TEST 212 RESERVED
; 3511 DUMMY NEWTST

```
:1 .NOLIST
:1804 .LIST
:1805
```

```
        :1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR56
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
                ;1841   .PAGE
                ;1842   .TOC " MICROTRAP HANLDE AND LSI ROUTINES"
                ;1843 ;*
                ;1844 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1845 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
                ;1846 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1847 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1848 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1849 ;
                ;1850 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1851 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1852 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1853 ; R[0F] AND DO A RETURN0.
                ;1854 ;
                ;1855 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1856 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1857 ; TO THE CONSOLE.
                ;1858 ;
                ;1859 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
                ;1860 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
                ;1861 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
                ;1862 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
                ;1863 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
                ;1864 ;-
                ;1865
U 1100. 0000.003C.19C0.0A78.0084.7001   ;1866 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101. 0000.003C.05C0.0A78.0084.7001   ;1867 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102. 0000.003C.09C0.0A78.0084.7001   ;1868 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103. 0000.003C.0DC0.0A78.0084.7001   ;1869 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104. 0000.003C.11C0.0A78.0084.7001   ;1870 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105. 0000.003C.D5C0.0A78.0084.7001   ;1871 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106. 0000.003C.D9C0.0A78.0084.7001   ;1872 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107. 0000.003C.5DC0.0A78.0084.7001   ;1873 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108. 0000.003C.01C0.0A78.0084.7001   ;1874 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C. 0000.003C.85C0.0A78.0084.7001   ;1875 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D. 0000.003C.89C0.0A78.0084.7001   ;1876 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E. 0000.003C.65C0.0A78.0084.7001   ;1877 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F. 0000.003C.61C0.0A78.0084.70E8   ;1878 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
                ;1879
U 1001. 0001.2024.0180.0800.0010.1007   ;1880 TRPH0: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1007. 0000.013C.0180.0800.0000.1002   ;1881  Z?    ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000   ;1882 =0 ALU_Q(AND)MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1883
                ;1884
                ;1885 ;*
                ;1886 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1887 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1888 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1889 ;
                ;1890 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1891 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1892 ;       FAULT STATUS REGISTER
                ;1893 ;       SBI ERROR REGISTER
                ;1894 ;       TIMEOUT ADDRESS REGISTER
                ;1895 ;       MAINTENANCE REGISTER
```

```
            ;1896 ;       CACHE PARITY REGISTER
            ;1897 ;       TB ERROR REGISTER 1
            ;1898 ;       TB ERROR REGISTER 0
            ;1899 ;-
            ;1900
U 1003, 0018.0038.1D80.09E8.0000.1024  ;1901 TRPH1: RC[0D]_SC
U 1024, 0000.003D.D980.0800.0084.7134  ;1902 =0 SETRCF[.9]
U 1025, 0000.003C.49F0.2C00.0000.1013  ;1903  Q_ID[TBER0]
U 1013, 0001.203C.4DF0.2DB0.0000.1017  ;1904  RC[6]_Q,Q_ID[TBER1]
U 1017, 0001.203C.65F0.2DB8.0000.1026  ;1905  RC[7]_Q,Q_ID[SBI.ERR]
U 1026, 0001.203C.6DF0.2DD8.0000.102E  ;1906  RC[0B]_Q,Q_ID[FAULT]
U 102E, 0001.203C.75F0.2DE0.0000.104F  ;1907  RC[0C]_Q,Q_ID[MAINT]
U 104F, 0001.203C.79F0.2DC8.0000.105B  ;1908  RC[9]_Q,Q_ID[PARITY]
U 105B, 0001.203C.69F0.2DC0.0000.105F  ;1909  RC[8]_Q,Q_ID[TIME.ADDR]
U 105F, 0001.203C.81F0.2DD0.0000.1000  ;1910  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1109  ;1911 1000: RC[0E]_Q,ERROR1
            ;1912
            ;1913 ;*
            ;1914 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
            ;1915 ;-
            ;1916
U 10E8, 0001.2024.0180.0800.0010.10E9  ;1917 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 10E9, 0000.013C.0180.0800.0000.102C  ;1918  Z?     ; BRANCH IF NO
U 102C, 0001.2036.0180.0AF8.0000.0001  ;1919 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1  ; RETURN
U 102D, 0000.003C.0180.0800.0000.1003  ;1920  J/TRPH1     ; REPORT UNEXPECTED TRAP
            ;1921
            ;1922
            ;1923 ;*
            ;1924 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1925 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1926 ;  NEWTST
            ;1927 ;  ERLOOP
            ;1928 ;  ERROR1
            ;1929 ;  ERROR2
            ;1930 ;
```

```
                ;1931  .PAGE
                ;1932  .TOC " Micro Monitor Interface Routines"
                ;1933  .TOC "   Newtest Routine"
                ;1934 ;++
                ;1935 ; FUNCTIONAL DESCRIPTION:
                ;1936 ;
                ;1937 ; This routine Initializes the following registers:
                ;1938 ;  PSL   to 1F
                ;1939 ;  CES   to 0
                ;1940 ;  ACC.CS   to disable accellerator
                ;1941 ;  SIR   to 0
                ;1942 ;  CLK.CS   to stop interval timer
                ;1943 ;  TBER0   to disable the TB
                ;1944 ;  SBI.MAINT to    0
                ;1945 ;  SBI.ERR   to 0
                ;1946 ;  FAULT   to 0
                ;1947 ;  COMP   to 0
                ;1948 ;  RC[0E]   to  0
                ;1949 ;  R[0F]   to 0
                ;1950 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1951 ; A SCOPE call is then made to the Monitor.
                ;1952 ;
                ;1953 ; CALLING CONSTRAINT:
                ;1954 ;
                ;1955 ; =0
                ;1956 ;
                ;1957 ; SIDE EFFECTS:
                ;1958 ;
                ;1959 ; D Reg is destroyed
                ;1960 ; SC is destroyed
                ;1961 ;--
U 10EA, 0000.003C.65F8.0800.0084.70EB  ;1962 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 10EB, 0818.0038.8D80.0800.0000.10EC  ;1963  d_k[.1f]  ; D=1F
U 10EC, 0D00.003C.0180.0800.0000.10ED  ;1964  d_dai.sc  ; D=001F0000
U 10ED, 0000.003C.3D80.3C00.0000.10EE  ;1965  id[psl]_d  ; psl = 001F0000
U 10EE, 0818.0038.0580.0800.0000.10EF  ;1966  d_k[.1]  ; Clear the CES register
U 10EF, 0D00.003C.0180.0800.0000.10F0  ;1967  d_dai.sc  ; D = 00010000
U 10F0, 0F00.003C.3180.3C00.0000.10F1  ;1968  id[ces]_d,d_0  ; ces = 00010000
U 10F1, 0000.003C.5D80.3C00.0000.10F2  ;1969  id[acc.cs]_d  ; acc.cs = 0
U 10F2, 0000.003C.3980.3C00.0000.10F3  ;1970  id[sir]_d  ; sir = 0
U 10F3, 0000.003C.2980.3C00.0000.10F4  ;1971  id[clk.cs]_d  ; clk.cs = 0
U 10F4, 0000.003C.4980.3C00.0000.1034  ;1972  id[tber0]_d  ; tber0 = 0
U 1034, 0000.003D.0180.0800.0000.1144  ;1973 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1974  intrpt.strobe,  ; Strobe interrupts
U 1035, 0818.0038.C180.0800.4000.10F5  ;1975      d_k[.ffff]  ; Setup to clear SBI error register and
U 10F5, 0801.0028.6580.3C00.0000.10F6  ;1976  id[sbi.err]_d,d_not.d ; and fault register
U 10F6, 0F00.003C.6D80.3C00.0000.10F7  ;1977  id[fault]_d,d_0  ; ...
U 10F7, 0000.003C.6D80.3C00.0000.10F8  ;1978  id[fault]_d  ; fault = 0
U 10F8, 0000.003C.7580.3C00.0000.10F9  ;1979  id[maint]_d  ; MAINT = 0
U 10F9, 0000.003C.6580.3C00.0000.10FA  ;1980  id[sbi.err]_d  ; sbi error reg = 0
U 10FA, 0000.003C.7180.3C00.0000.10FB  ;1981  id[comp]_d  ; comp reg = 0
U 10FB, 0000.003C.E580.3C00.0000.10FC  ;1982  id[T9]_d  ; Clear code for subroutine START.TEST
U 10FC, 0001.003C.0180.09F0.0000.10FD  ;1983  rc[0e]_d  ;
U 10FD, 0001.003C.0180.0AF8.0000.10FE  ;1984  r[0f]_d  ; Clear expected trap mask
U 10FE, 0001.003C.0180.09F8.0000.1036  ;1985  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1036, 0000.003D.0180.0800.0000.1114  ;1986 =0 call[disable.cache] ; Disable the cache
U 1037, 0F03.003C.0180.09E8.0000.105E  ;1987  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1988
U 10FF, 0818.0038.0980.0800.0000.105E  ;1989 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
          ;1990
```

```
            ;1991   .PAGE
            ;1992   .TOC "    ERROR1 WITH PARAMETER"
            ;1993 ;++
            ;1994 ; FUNCTIONAL DESCRIPTION:
            ;1995 ;
            ;1996 ; This subroutine takes  as parameter the number of arguments
            ;1997 ; to be printed to the console in the case of an error logout.
            ;1998 ;
            ;1999 ; CALLING CONSTRAINT:
            ;2000 ;
            ;2001 ; =0
            ;2002 ;INPUTS:
            ;2003 ;
            ;2004 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2005 ;--
            ;2006 =0
            ;2007 ERR1.ARG:
U 1038, 0000,003D,0180,0800,0000,1134  ;2008  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1039, 0000,003C,0180,0800,0000,1109  ;2009  J/ERROR1  ; Go to ERROR1
            ;2010 ;
            ;2011 ;
U 1109, 0810,0038,0180,0978,0000,110A  ;2012 ERROR1: D_RC[0F]
U 110A, 0819,0034,4D80,0800,0000,104E  ;2013  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2014 104E: D_D.OR.K[.4],J/CALLCON
            ;2015
```

```
            ;2016  .PAGE
            ;2017  .TOC "   ERROR 2 WITH PARAMETER"
            ;2018 ;++
            ;2019 ; FUNCTIONAL DESCRIPTION:
            ;2020 ;
            ;2021 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2022 ; except that in this case after setting the number of arguments
            ;2023 ; too the console, control is transferred to routine ERROR2.
            ;2024 ;
            ;2025 ; INPUTS:
            ;2026 ;
            ;2027 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2028 ;
            ;2029 ;--
            ;2030 =0
            ;2031 ERR2.ARG:
U 103A, 0000,003D,0180,0800,0000,1134  ;2032  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 103B, 0000,003C,0180,0800,0000,110B  ;2033  J/ERROR2  ; Go to ERROR2
            ;2034 ;
            ;2035 ;
U 110B, 0810,0038,0180,0978,0000,1110  ;2036 ERROR2: D_RC[0F]
U 1110, 0819,0034,4D80,0800,0000,105A  ;2037  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2038 105A: D_D.OR.K[.6],J/CALLCON
            ;2039
            ;2040 105E:
            ;2041 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2042  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2043
            ;2044 ;*
            ;2045 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2046 ;-
            ;2047
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2048 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2049 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2050 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2051 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2052 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2053 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2054 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2055 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2056 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2057 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2058 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2059 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2060 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2061 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2062 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1111  ;2063 13FF: NOP
            ;2064
```

```
            ;2065  .PAGE
            ;2066  .TOC "    Set Trap Mask"
            ;2067 ;**
            ;2068 ; FUNCTIONAL DESCRIPTION:
            ;2069 ;
            ;2070 ; This routine is used to set a bit in the Micro Trap Mask
            ;2071 ; R[0F].
            ;2072 ;
            ;2073 ; CALLING CONSTRAINT:
            ;2074 ;
            ;2075 ; =0
            ;2076 ;
            ;2077 ; INPUTS:
            ;2078 ;
            ;2079 ; SC - Contains bit number to set.
            ;2080 ;
            ;2081 ; OUTPUTS:
            ;2082 ;
            ;2083 ; rc[0E] - Contains the current trap mask
            ;2084 ;
            ;2085 ; SIDE EFFECTS:
            ;2086 ;
            ;2087 ; d regsiter is destroyed
            ;2088 ;--
            ;2089 SET_TRAP_MASK:
U 1111, 0800,003C,0180,0A78,0000,1112  ;2090  d_r[0f]
U 1112, 0801,001C,0180,0AF8,0000,1113  ;2091  r[0f]_d.ornot.mask,d_alu ; Set mask
U 1113, 0001,003E,0180,0AF0,0000,0001  ;2092  r[0E]_d,return1    ; Save mask and return
            ;2093
```

```
            ;2094  .PAGE
            ;2095  .TOC "   Disable Cache Routine"
            ;2096 ;**
            ;2097 ; FUNCTIONAL DESCRIPTION:
            ;2098 ;
            ;2099 ; This routine disables cache hits by setting bits <16:15>
            ;2100 ; in the maintenance register.
            ;2101 ;
            ;2102 ; CALLING SEQUENCE:
            ;2103 ;
            ;2104 ; =0
            ;2105 ;
            ;2106 ; SIDE EFFECTS:
            ;2107 ;
            ;2108 ; D & Q Registers destroyed
            ;2109 ;--
            ;2110 DISABLE.CACHE:
U 1114, 0818,0038,4580,0800,0000,1115  ;2111  d_k[.8000]  ; ... and seed constant
U 1115, 0500,003C,0180,0800,0000,1116  ;2112  d_d.left  ; Generate final constant
U 1116, 0819,0030,4580,0800,0000,1117  ;2113  d_d.or.k[.8000]  ; ... = 00018000
U 1117, 0000,003E,7580,3C00,0000,0001  ;2114  id[maint]_d,return1 ; Disable cache and return
            ;2115
```

```
                ;2116  .PAGE
                ;2117  .TOC "   Initialize the SBI"
                ;2118 ;**
                ;2119 ; FUNCTIONAL DESCRIPTION:
                ;2120 ;
                ;2121 ; This routine performs the following functions:
                ;2122 ;
                ;2123 ;  Executes an SBI Unjam
                ;2124 ;  Clears the SBI Fault Latch
                ;2125 ;  Clears the SBI Error Register
                ;2126 ;
                ;2127 ; CALLING CONSTRAINT:
                ;2128 ;
                ;2129 ; =0
                ;2130 ;
                ;2131 ; SIDE EFFECTS:
                ;2132 ;
                ;2133 ; D & Q Register destroyed
                ;2134 ;--
                ;2135 =0
                ;2136 SBI.INIT:
U 103E, 0000.003D.0180.0800.0000.1144  ;2137  call[sbi.unjam]  ; Unjam the SBI
U 103F, 0818.0038.C180.0800.0000.1118  ;2138  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1118, 0801.0028.6580.3C00.0000.1119  ;2139  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1119, 0F00.003C.6D80.3C00.0000.111A  ;2140  id[fault]_d,d_0
U 111A, 0000.003C.6D80.3C00.0000.111B  ;2141  id[fault]_d
U 111B, 0000.003E.6580.3C00.0000.0001  ;2142  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
                ;2143
                ;2144
                ;2145  .TOC "   ASCII MESSAGES"
                ;2146 ;*
                ;2147 ; FOLLOWING ARE THE ASCII MESSAGES USED BY THIS MODULE
                ;2148 ;
                ;2149 ;& 1,/TEST ABORTED - NOT ENOUGH MEMORY/
                ;2150 ;& 2,/TEST ABORTED - NO DW780/
                ;2151 ;
                ;2152
                ;2153
                ;2154 ;*
                ;2155 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
                ;2156 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
                ;2157 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
                ;2158 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
                ;2159 ;
                ;2160 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
                ;2161 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
                ;2162 ;
                ;2163 ;   =0 D_0,CALL,J/DC5
                ;2164 ;    NOP
                ;2165 ;   =0 CALL,J/DC5
                ;2166 ;-
                ;2167
U 111C, 0018.0038.1980.0800.0000.112C  ;2168 DC0: ALU_0(K),J/S00
U 111D, 0018.0038.1980.0800.0000.112D  ;2169 DC1: ALU_0(K),J/S01
U 111E, 0018.0038.1980.0800.0000.112E  ;2170 DC2: ALU_0(K),J/S10
```

```
U 111F, 0018,0038,1980,0800,0000,112F  ;2171 DC3: ALU_0(K),J/S11
U 1120, 0018,0038,0980,0800,0000,112C  ;2172 DC4: ALU_K[.2],J/S00
U 1121, 0018,0038,0980,0800,0000,112D  ;2173 DC5: ALU_K[.2],J/S01
U 1122, 0018,0038,0980,0800,0000,112E  ;2174 DC6: ALU_K[.2],J/S10
U 1123, 0018,0038,0980,0800,0000,112F  ;2175 DC7: ALU_K[.2],J/S11
U 1124, 0018,0038,0580,0800,0000,112C  ;2176 DC8: ALU_K[.1],J/S00
U 1125, 0018,0038,0580,0800,0000,112D  ;2177 DC9: ALU_K[.1],J/S01
U 1126, 0018,0038,0580,0800,0000,112E  ;2178 DCA: ALU_K[.1],J/S10
U 1127, 0018,0038,0580,0800,0000,112F  ;2179 DCB: ALU_K[.1],J/S11
U 1128, 0018,0038,C180,0800,0000,112C  ;2180 DCC: ALU_K[.FFFF],J/S00
U 1129, 0018,0038,C180,0800,0000,112D  ;2181 DCD: ALU_K[.FFFF],J/S01
U 112A, 0018,0038,C180,0800,0000,112E  ;2182 DCE: ALU_K[.FFFF],J/S10
U 112B, 0018,0038,C180,0800,0000,112F  ;2183 DCF: ALU_K[.FFFF],J/S11
          ;2184
U 112C, 0118,0038,1B80,0800,0000,1130  ;2185 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 112D, 0118,0038,0B80,0800,0000,1130  ;2186 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 112E, 0118,0038,0780,0800,0000,1130  ;2187 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 112F, 0118,0038,C380,0800,0000,1130  ;2188 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
          ;2189
U 1130, 0100,003E,0380,0800,0000,0001  ;2190 SX: D_D.LEFT2,SI/7,RETURN1  ; EXIT
          ;2191
          ;2192
          ;2193 ;+
          ;2194 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
          ;2195 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
          ;2196 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
          ;2197 ; TYPING IT.
          ;2198 ;-
          ;2199
U 1131, 0810,0038,4D80,0978,0000,1132  ;2200 SUBTST: D_RC[0F],KMX/.FF00
U 1132, 0019,4016,4D80,09F8,0000,0001  ;2201  RC[0F]_D+K[.FF00],WORD,RETURN1
          ;2202
```

```
            ;2203  .PAGE
            ;2204  .TOC "   RESET SUBTEST NUMBER"
            ;2205 ;**
            ;2206 ; FUNCTIONAL DESCRIPTION:
            ;2207 ;
            ;2208 ; Since some  of the tests will have to be restarted when two
            ;2209 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2210 ; the subtest counter to zero ( only for thoes tests that have
            ;2211 ; more than one subtest). This subroutine may also be necessary
            ;2212 ; when a test is being loop on.
            ;2213 ;
            ;2214 ; CALLING CONSTRAINT:
            ;2215 ;
            ;2216 ; =0
            ;2217 ; SIDE EFFECTS:
            ;2218 ;
            ;2219 ; D is destroyed
            ;2220 ;
            ;2221 ;--
            ;2222 RESET.SUBTST:
            ;2223  RC[OF]_0,   ; Reset subtest number
U 1133, 0003,003E,0180,09F8,0000,0001  ;2224   RETURN1   ; ...and return
            ;2225
            ;2226
            ;2227 ;*
            ;2228 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2229 ; LOW BYTE OF RC[OF] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2230 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2231 ;-
            ;2232
U 1134, 0010,0038,01C0,0978,0000,1135  ;2233 SETRCF: Q_RC[OF]
U 1135, 0019,2034,4DC0,0800,0000,1136  ;2234  Q_Q.AND.K[.FF00]
U 1136, 0019,2032,1D80,09F8,0000,0001  ;2235  RC[OF]_Q.OR.SC,RETURN1
U 1155, 0019,2000,05C0,0800,0000,12AB  ;2236 1155: Q_Q-K[.1],J/12AB
U 12AA, 0000,003C,0180,0800,0000,1137  ;2237 12AA: NOP     ; ECO LOCATION
            ;2238
            ;2239
            ;2240 DISPATIRD:
U 1137, 0000,003F,0180,0800,1000,1200  ;2241  SUB/SPEC,MSC/IRD,J/1200
            ;2242
U 1220, 0000,003E,01F8,0800,0000,0001  ;2243 1220: Q_0,RETURN1
U 1224, 0018,003A,05C0,0800,0000,0001  ;2244 1224: Q_K[.1],RETURN1
U 1228, 0018,003A,09C0,0800,0000,0001  ;2245 1228: Q_K[.2],RETURN1
U 122C, 0018,003A,0DC0,0800,0000,0001  ;2246 122C: Q_K[.3],RETURN1
U 1230, 0018,003A,11C0,0800,0000,0001  ;2247 1230: Q_K[.4],RETURN1
U 1234, 0018,003A,D5C0,0800,0000,0001  ;2248 1234: Q_K[.6],RETURN1
U 1238, 0018,003A,5DC0,0800,0000,0001  ;2249 1238: Q_K[.7],RETURN1
U 123C, 0018,003A,01C0,0800,0000,0001  ;2250 123C: Q_K[.8],RETURN1
            ;2251
            ;2252
            ;2253
            ;2254 ;*
            ;2255 ; THE FOLLOWING ROUTINES ARE USES TO SET OR CLEAR BITS IN THE CONSOLE
            ;2256 ; REGISTERS. BYTE 0 OF THE D REGISTER MUST CONTAIN "C", BYTE ONE OF THE
            ;2257 ; D REGISTER MUST CONTAIN THE CONSOLE REGISTER ID, I.E. 0=TXCS, 2=RXCS,
```

```
            ;2258 ; AND 4=MCR, AND THE UPPER WORD OF THE D REGISTER MUST CONTAIN THE DATA
            ;2259 ; TO SET IN THE REGISTER.
            ;2260 ;
            ;2261 ; THE SUBROUTINES MUST BE CALLED WITH THE DATA TO LOAD INTO THE REGISTER
            ;2262 ; IN SCRATCH PAD R[0A].
            ;2263 ;
            ;2264 ; NOTE: ONLY BIT 15, 9, AND 8 OF THE MCR REGISTER CAN BE SET OR CLEARED.
            ;2265 ;       ONLY BIT 7 OF THE RXCS REGISTER CAN BE SET OR CLEARED.
            ;2266 ;       BIT 7 OF THE TXCS REGISTER IS ALWAYS SET WHILE THE MICRO CODE IS RUNNING.
            ;2267 ;-
            ;2268
U 1138, 0818,0038,1180,0800,0000,1139  ;2269 SETMCR: D_K[.4]     ; GET REGISTER ID
U 1139, 0000,003C,01F8,0800,0084,713A  ;2270  SC_K[.8],Q_0     ; SETUP FOR SHIFT
U 113A, 0D00,003C,0180,0800,0000,113F  ;2271  D_DAL.SC,J/COMCON   ; SHIFT TO BYTE 1
            ;2272
U 113B, 0F00,003C,01F8,0800,0000,113F  ;2273 SETTXCS:D_0,J/COMCON,Q_0   ; SET REGISTER ID
U 113C, 0818,0038,0980,0800,0000,113D  ;2274 SETRXCS:D_K[.2]     ; GET REGISTER ID
U 113D, 0000,003C,01F8,0800,0084,713E  ;2275  SC_K[.8],Q_0     ; SETUP FOR SHIFT
U 113E, 0D00,003C,0180,0800,0000,113F  ;2276  D_DAL.SC,J/COMCON   ; PUT IN BYTE 1
            ;2277
U 113F, 0019,0030,8580,0AD8,0000,1140  ;2278 COMCON: R[0B]_D.OR.K[.C]   ; INSERT BYTE 0 AND SAVE
U 1140, 0800,003C,0180,0A50,0000,1141  ;2279  D_R[0A]      ; GET DATA TO PUT IN CONSOLE REG
U 1141, 0000,003C,6580,0800,0084,7142  ;2280  SC_K[.10]     ; SETUP FOR SHIFT
U 1142, 0D00,003C,0180,0A58,0000,1143  ;2281  D_DAL.SC,LAB_R[0B]   ; PUT IN UPPER WORD
U 1143, 081C,2030,0180,0800,0000,105E  ;2282  D_LA.OR.D,J/CALLCON   ; INSERT LOWER WORD AND CALL CONSOLE
            ;2283
            ;2284
            ;2285
            ;2286
            ;2287 ;*
            ;2288 ; THIS SUBROUTINE EXECUTE AN SBI UNJAM SEQUENCE
            ;2289 ;-
            ;2290
            ;2291 SBI.UNJAM:
U 1144, 0F00,003C,0180,0800,0000,1145  ;2292 UNJAM: D_0
U 1145, 0000,003C,7580,3C00,0000,1146  ;2293  ID[MAINT]_D     ; RELOAD THE MAT REG
U 1146, 0F00,003C,0180,8000,0000,1147  ;2294  D_0,MCT/SBI.HOLD   ; ASSERT HOLD FOR 16 CYCLES
U 1147, 0019,0020,6180,8000,0010,1148  ;2295 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD ; DONE YET?
U 1148, 0000,013C,0180,8000,0000,1040  ;2296  Z?,MCT/SBI.HOLD    ; BRANCH IF YES
U 1040, 0819,0014,0580,8000,0000,1147  ;2297 =0 D_D*K[.1],MCT/SBI.HOLD,J/UNJAM0 ; CONTINUE HOLDING THE BUS
            ;2298
U 1041, 0F00,003C,0180,8800,0000,1149  ;2299  D_0,MCT/SBI.HOLD*UNJAM   ; ASSERT HOLD AND UNJAM FOR 16 CYCLES
U 1149, 0019,0020,6180,8800,0010,114A  ;2300 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,MCT/SBI.HOLD*UNJAM ; DONE YET?
U 114A, 0000,013C,0180,8800,0000,1042  ;2301  Z?,MCT/SBI.HOLD*UNJAM   ; BRANCH IF YES
U 1042, 0819,0014,0580,8800,0000,1149  ;2302 =0 D_D*K[.1],MCT/SBI.HOLD*UNJAM,J/UNJAM1 ; CONTINUE UNJAM
            ;2303
U 1043, 0F00,003C,0180,8000,8000,114B  ;2304  D_0,MCT/SBI.HOLD,INTRPT.ACK ; ASSERT HOLD FOR 16 CYCLES
U 114B, 0019,0020,6180,8000,0010,114C  ;2305 UNJAM2: ALU_D[XOR]K[.F],MCT/SBI.HOLD,CLK.UBCC ; DONE YET?
U 114C, 0000,013C,0180,8000,0000,1044  ;2306  Z?,MCT/SBI.HOLD    ; BRANCH IF YES
U 1044, 0819,0014,0580,8000,0000,114B  ;2307 =0 D_D*K[.1],MCT/SBI.HOLD,J/UNJAM2 ; CONTINUE HOLDING
            ;2308
U 1045, 0000,003C,8580,0800,0084,714D  ;2309  SC_K[.C]
U 114D, 0803,001C,0180,0800,4000,114E  ;2310  D_NOT.MASK,INTRPT.STROBE
U 114E, 0000,003E,7580,3C00,0000,0001  ;2311  ID[MAINT]_D,RETURN1   ; TURN OFF THE SBI
            ;2312
```

```
            ;2313
            ;2314
            ;2315 ;+
            ;2316 ; THE FOLLOWING SUBROUTINE GENERATES CONSTANTS AND SAVES THEM IN THE RA
            ;2317 ; SCRATCH PADS. FOLLOWING ARE THE CONSTANTS GENERATED:
            ;2318 ;
            ;2319 ; R[2] = FORCE REPLACE GROUP 1 AND FORCE MISS GROUP 0
            ;2320 ; R[3] = R[6] PLUS DISABLE SBI CYCLE
            ;2321 ; R[4] = CPU ADDRESS FOR AUTO REFILL (0)
            ;2322 ; R[5] = IB ADDRESS FOR AUTO REFILL (3FC)
            ;2323 ; R[6] = FORCE REPLACE GROUP 0 AND FORCE MISS GROUP 1
            ;2324 ; R[7] = GROUP 0 AND GROUP 1 MATCH BIT MASK
            ;2325 ; R[8] = GROUP 0 MATCH BIT
            ;2326 ; R[9] = GROUP 1 MATCH BIT
            ;2327 ; R[A] = R[6] PLUS REVERSE PARITY GROUP 0, BYTE 0 (DATA)
            ;2328 ; R[B] = TIMEOUT ADDRESS ( = 01000000 )
            ;2329 ; R[C] = SILO COUNTER MASK
            ;2330 ; R[D] = ADDRESS TO FORCE RDS (4020)
            ;2331 ; R[E] = ADDRESS OF MEMORY CONFIGURATION REGISTER B
            ;2332 ;-
            ;2333
            ;2334 =0
            ;2335 MAINTREGSETUP:
U 1046, 0000,003D,F580,0800,0084,7134  ;2336  SETRCF[.A]    ; Set the Argument count to CONSOLE
U 1047, 0000,003C,0180,0800,0000,1048  ;2337  NOP
U 1048, 0000,003D,0180,0800,0000,1170  ;2338 =0 CALL[CONFIG.MS780E]  ; Find MS780E memory and configure it
U 1049, 0810,0038,0180,0970,0000,114F  ;2339  D_RC[0E]    ; Get CNFG Reg A Address
U 114F, 0019,0014,1180,0AF0,0000,1150  ;2340  R[0E]_D+K[.4]    ; Make it in CNFG B Address
U 1150, 0818,0038,8580,0800,0000,1151  ;2341  D_K[.C]     ; Form "FORCE RDS" constant for
            ;2342       ; ...CNFG Reg B in R[0D]
            ;2343       ; ...First ECC substitute bits for
            ;2344       ; ...data of zero
U 1151, 0000,003C,F580,0800,0084,7152  ;2345  SC_K[.A]    ; And now the Diagnostic Mode bits
            ;2346  D_D.ORNOT.MASK.    ; ...and save them in
U 1152, 0801,001C,0180,0AE8,0000,104A  ;2347   R[0D]_ALU  ; ...R[0D]
U 104A, 0000,003D,0180,0800,0000,1144  ;2348 =0 CALL[SBI.UNJAM]    ;
U 104B, 0000,003C,0180,0800,0000,1153  ;2349  NOP
U 1153, 0003,003C,0180,0AA0,0000,104C  ;2350  R[4]_0
U 104C, 0818,0039,5580,0800,0000,1128  ;2351 =0 D_K[.3F],CALL,J/DCC  ; GENERATE 3FC
U 104D, 0001,003C,0180,0AA8,0000,1050  ;2352  R[5]_D     ; SAVE
U 1050, 0818,0039,7580,0800,0000,111C  ;2353 =0 D_K[.20],CALL,J/DC0  ; GENERATE 200(X)
U 1051, 0001,003C,0180,0AC0,0000,1154  ;2354  R[8]_D     ; GROUP 0 MATCH BIT
U 1154, 0501,003C,0180,0AB8,0000,1156  ;2355  R[7]_D,D_D.LEFT    ; PART OF MASK
U 1156, 0001,003C,0180,0AC8,0000,1157  ;2356  R[9]_D     ; GROUP 1 MATCH BIT
U 1157, 0000,003C,0180,0A38,0000,1158  ;2357  LAB_R[7]
U 1158, 001C,2030,0180,0AB8,0000,1159  ;2358  R[7]_LA.OR.D    ; COMPLETE THE MASK
U 1159, 0100,003C,0180,0800,0000,115A  ;2359  D_D.LEFT2
U 115A, 0500,003C,0180,0800,0000,115B  ;2360  D_D.LEFT    ; GENERATE FR GR0 AND FM GR1
U 115B, 0501,003C,0180,0A90,0000,115C  ;2361  R[2]_D,D_D.LEFT
U 115C, 0500,003C,01E0,0800,0000,115D  ;2362  Q_D,D_D.LEFT    ; ...
U 115D, 081D,0030,01E0,0AB0,0000,115E  ;2363  R[6]_D.OR.Q,D_ALU,Q_D   ; SAVE
U 115E, 0000,003C,7580,3C00,0000,115F  ;2364  ID[MAINT]_D    ; SETUP THE MAINTENANCE REGISTER
U 115F, 0000,003C,8580,0800,0084,7160  ;2365  SC_K[.C]    ; SETUP TO GENERATE BIT 12
U 1160, 0801,001C,0180,0800,0000,1161  ;2366  D_D.ORNOT.MASK    ; INSERT BIT 12
U 1161, 0001,003C,0180,0A98,0000,1162  ;2367  R[3]_D     ; SAVE
```

```
U 1162, 0C00.003C.01B0.0A10.0000.1163  ;2368  LAB_R[2],D_Q    ; COMPLETE DATA FOR R2
U 1163, 0500.003C.01B0.0800.0000.1164  ;2369  D_D.LEFT
U 1164, 001C.2030.01B0.0A90.0000.1165  ;2370  R[2]_LA.OR.D    ;
U 1165, 0818.0038.51B0.0800.0000.1166  ;2371  D_K[.1E]     ; GENERATE CONSTANT FOR R[A]
U 1166, 0000.003C.65F8.0800.0084.7167  ;2372  SC_K[.10],Q_0    ; ...
U 1167, 0D00.003C.01C0.0A30.0000.1168  ;2373  D_DAL.SC,Q_R[6]    ; ...
U 1168, 001D.0030.01B0.0AD0.0000.1169  ;2374  R[0A]_D.OR.Q    ; SAVE
U 1169, 0818.0038.05B0.0800.0000.116A  ;2375  D_K[.1]     ; GENERATE A TIMEOUT ADDRESS
U 116A, 0B00.003C.01B0.0800.0000.116B  ;2376  D_D.SWAP    ; ...
U 116B, 0001.003C.01B0.0AD8.0000.116C  ;2377  R[0B]_D     ; SAVE
U 116C, 0818.0038.61F8.0800.0000.116D  ;2378  D_K[.F],Q_0
U 116D, 0D00.003C.01B0.0800.0000.116E  ;2379  D_DAL.SC
U 116E, 0001.003C.01B0.0AE0.0000.116F  ;2380  R[0C]_D     ; SILO COUNTER MASK
U 116F, 0000.003E.01B0.0800.0000.0001  ;2381  RETURN1
            ;2382
            ;2383
```

```
;2384  .PAGE
;2385  .TOC "   CONFIGURE MS780-E/H"
;2386 ;*************************************************************
;2387 ;++
;2388 ;
;2389 ; Functional Description:
;2390 ; ----------------------
;2391 ;
;2392 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2393 ; specifically  check  the  memory, but  make use of it to execute.
;2394 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2395 ; that  if  a  Read/Write operation will take place no false errors
;2396 ; will be logged due to misconfigured memory.
;2397 ;
;2398 ; Calling Constraint: =00
;2399 ; ------------------
;2400 ;
;2401 ;
;2402 ; Inputs:
;2403 ; ------
;2404 ;
;2405 ; RC[0E] - I/O BASE OF MS780-E/H
;2406 ;
;2407 ;
;2408 ; Outputs:
;2409 ; -------
;2410 ;
;2411 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2412 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2413 ;      OR IF NO MS780-E/Hs WERE FOUND
;2414 ;      ON THE SBI
;2415 ;
;2416 ; Side Effects:
;2417 ; ------------
;2418 ;
;2419 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2420 ;
;2421 ;
;2422 ;
;2423 ;--
;2424 ;*************************************************************
;2425 CONFIG.MS780E:
U 1170, 0818.0038.0D80.0800.0000.1171  ;2426  D_K[.3]     ; Set up flag for the Fail Chain
U 1171, 0001.003C.0180.09E8.0000.1004  ;2427  RC[0D]_D     ; ...
U 1004, 0000.003D.0180.0800.0000.1052  ;2428 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1005, 0000.003D.0980.0800.0084.7038  ;2429  ERROR1[.2]    ; No MS780-E/H's found on the SBI
U 1006, 0000.003C.0180.0800.0000.1010  ;2430  NOP      ; OK. Found at least one MS780-E/H
;2431 =
;2432 CONFIG.RETRY:     ; Entry point for the case in which the
;2433       ; ...first MS780-E/H could not be
;2434       ; ...properly configured
U 1010, 0000.003D.0180.0800.0000.1028  ;2435 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1011, 0000.003C.0180.0800.0000.1014  ;2436  J/CNFG.LMIS    ; Lower controller misconfigured
U 1012, 0000.003E.0180.0800.0000.0001  ;2437  RETURN1     ; OK. Lower Controller is configured
;2438 =
```

```
            ;2439 =00
            ;2440 CNFG.LMIS:
U 1014, 0000,003D,0180,0800,0000,1030  ;2441  CALL[SELECT.UPPER]  ; Select the upper controller
U 1015, 0000,003C,0180,0800,0000,1020  ;2442  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 1016, 0000,003E,0180,0800,0000,0001  ;2443  RETURN1    ; OK. Upper Controller is configured
            ;2444 =
            ;2445 =00
            ;2446 CNFG.UMIS:
U 1920, 0000,003D,0180,0800,0000,11A5  ;2447  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2448       ; ...MS780-E/H found so go and see
            ;2449       ; ...if there are any more
U 1021, 0000,003C,0180,0800,0000,1172  ;2450  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2451       ; ...try to configure it
U 1022, 0818,0038,0D80,0800,0000,1023  ;2452  D_K[.3]    ; Set Flag for Fail Chain
U 1023, 0001,003C,0180,09E8,0000,1172  ;2453  RC[0D]_D    ; ...
U 1172, 0000,003D,0980,0800,0084,7038  ;2454  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2455       ; ...abort the test
            ;2456 =
            ;2457
```

```
            ;2458  .PAGE
            ;2459  .TOC "   Find MS780-E Memory"
            ;2460 ;**
            ;2461 ; FUNCTIONAL DESCRIPTION:
            ;2462 ;
            ;2463 ;          The diagnostic is designed to handle up to 4 (four)
            ;2464 ;     MS780-E/H's on the 780 SBI.  This routine looks at all TR levels
            ;2465 ;     for a MS780-E memory unit.  Upon return, ID registers 3A through
            ;2466 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2467 ;     on the SBI, and ID Register 3E will hold the Total number of
            ;2468 ;     MS780-E/H's found minus one.  Also, it is to be noted that the
            ;2469 ;     first MS780-E found will have its starting address set to 0
            ;2470 ;     (zero).
            ;2471 ;     All the other MS780-E/H's found will have their starting address
            ;2472 ;     set to all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2473 ;     Reg 3E to decide if there are any more MS780-E and will take
            ;2474 ;     apropriate action.  Register RC[0E] is used as a pointer to the
            ;2475 ;     current MS780-E unit under test.
            ;2476 ;     If any of:  MS780-A, MS780-C or MA780 is found, its address is
            ;2477 ;     set to maximum.
            ;2478 ;
            ;2479 ; CALLING CONSTRAINT:
            ;2480 ;
            ;2481 ; =00
            ;2482 ;
            ;2483 ; OUTPUTS:
            ;2484 ;
            ;2485 ; rc[0e] - Contains address of Configuration Register
            ;2486 ; r[0]  - Contains TR level
            ;2487 ;
            ;2488 ; SIDE EFFECTS:
            ;2489 ;
            ;2490 ; D register is destroyed.
            ;2491 ;--
            ;2492 =0
            ;2493 FIND.MS780E:
U 1052, 0000,003D,0180,0800,0000,103E  ;2494  call[sbi.init]  ; Make sure SBI is enabled
U 1053, 0000,003C,0180,0800,0000,1054  ;2495  NOP
U 1054, 0000,003D,0180,0800,0000,1114  ;2496 =0 CALL[DISABLE.CACHE] ; Disable the cache and re-enable the SBI
U 1055, 0003,003C,0180,0988,0000,1173  ;2497  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1173, 0818,0038,1980,0800,0000,1174  ;2498  d_k[zero]  ; Clear MS780-E/H counter
U 1174, 0000,003C,F980,3C00,0000,1175  ;2499  id[3e]_d  ; ...
U 1175, 0018,0038,0580,0A80,0000,1176  ;2500  r[0]_k[.1]  ; Init TR counter
U 1176, 0F03,003C,0180,09F0,0000,1056  ;2501  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2502      ; ...CNFG A Reg
            ;2503 =0
            ;2504 FIND.MEM.LOOP:
U 1056, 0800,003D,0180,0A00,0000,119C  ;2505  d_r[0],call[generate.io]; Generate address of TR level
U 1057, 0001,003C,0180,09F0,0200,1058  ;2506  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1058, 0000,003D,8980,0800,0084,7111  ;2507 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2508  d[long]_cache.p, ; Read the specified tr level
U 1059, 0000,003C,0180,C970,0000,1177  ;2509   ic_rc[0e]  ; ...
U 1177, 0000,003C,0180,0A78,0010,1178  ;2510  alu_r[0f],clk.ubcc ; Check for no response
U 1178, 0001,013C,0180,0880,0082,105C  ;2511  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 105C, 0000,003C,0180,0800,0000,1179  ;2512 =0 j/check.adpt.code ; Check for a memory
```

```
U 105D, 0000,003C,0180,0800,0000,1198  ;2513  j/find.mem.lp1  ; Try next tr
        ;2514 CHECK.ADPT.CODE:
U 1179, 0000,003C,1D80,0800,1404,717A  ;2515  state_sc,via.kmx ; Setup to branch on bits<6:4>
U 117A, 0000,163C,0180,0800,0000,1008  ;2516  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,717B  ;2517 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,717C  ;2518  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,1198  ;2519  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,1198  ;2520  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7181  ;2521  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,1198  ;2522  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1185  ;2523  rc[0e]_ic,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1185  ;2524  rc[0e]_ic,j/found.ms780e ; MS780-E
        ;2525 ;
        ;2526 ; Found an MS780-A or -C. Set the starting address
        ;2527 ; to maximum.
        ;2528 ;
U 117B, 0818,0038,5D80,0800,0000,117D  ;2529 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 117C, 0818,0038,0580,0800,0000,117D  ;2530 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
        ;2531 MS780.COMMON:
U 117D, 0819,0030,7180,0800,0000,117E  ;2532  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 117E, 0000,003C,01F8,0803,0080,D17F  ;2533  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 117F, 0D00,003C,0180,0800,0000,1180  ;2534  d_dal.sc  ; Put data in bits<29:14>
        ;2535  cache.p_d[long], ; Set the starting address to maximum
U 1180, 0000,003C,0180,A800,0000,1198  ;2536      j/find.mem.lp1 ; and continue TR scan
        ;2537 ;
        ;2538 ; Found an MA780. Set the starting address to maximum
        ;2539 ;
        ;2540 MA780.MAX:
U 1181, 0818,0038,39F8,0803,0000,1182  ;2541  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1182, 0D00,003C,0180,0803,0000,1183  ;2542  d_dal.sc,va_va+4 ; Put in proper position
U 1183, 0000,003C,0180,0803,0000,1184  ;2543  va_va+4   ; Point to Port Invalidate Ctrl Register
        ;2544  cache.p_d[long], ; Set starting address to maximum
U 1184, 0000,003C,0180,A800,0000,1198  ;2545      j/find.mem.lp1
        ;2546 ;
        ;2547 ; Found an MS780E
        ;2548 ;
        ;2549 FOUND.MS780E:
U 1185, 0000,003C,F9F0,2C00,0000,1186  ;2550  q_id[3e]  ; Set up to see how many MS780-E's
        ;2551  alu_q.xor.k[.3], ; Are there Maximum units?
U 1186, 0019,2020,0D80,0800,0010,1187  ;2552   clk.ubcc ; Check condition codes
U 1187, 0000,013C,0180,0800,0000,1060  ;2553  z?   ; Are we at maximum units for system
U 1060, 0000,003C,0180,0800,0000,1188  ;2554 =0 J/ms780e.tst  ; No go find how many units ther are
U 1061, 0818,0038,C180,0800,0000,1062  ;2555  d_k[.ffff]  ; Yes set address to maximum
U 1062, 0000,003D,0180,0800,0000,11A0  ;2556 =0 call[set.start.addr] ; .....
U 1063, 0000,003C,0180,0800,0000,1198  ;2557  j/find.mem.lp1  ; Go check to see if we are done
        ;2558 ;
        ;2559 MS780E.TST:
        ;2560  alu_rc[01],  ; Check "Found MS780E Flag"
U 1188, 0010,0038,0180,0908,0010,1189  ;2561   clk.ubcc ; Check condition codes
U 1189, 0810,0138,0180,0970,0000,1064  ;2562  z?,d_rc[0e]  ; Is this unit the first MS780E ?
        ;2563 =0 j/not.first.ms780e, ; No
U 1064, 0818,0038,C180,0800,0000,1068  ;2564   d_k[.ffff] ; Set starting address
U 1065, 0001,003C,0180,0988,0000,118A  ;2565  rc[01]_d  ; Yes put base address in rc[01]
U 118A, 0818,0038,7980,0800,0000,118B  ;2566  d_k[.30]  ; Set up SC to point to first unit
U 118B, 0019,0014,F580,0800,0082,118C  ;2567  alu_d+k[.a],sc_alu ; ...
```

```
U 118C. 0810.0038.0180.0908.0000.118D  ;2568  d_rc[01]  ; Put base address of unit in D
U 118D. 0000.003C.0180.3400.0000.118E  ;2569  id(sc)_d  ; Put base address in ID pointed to by SC
U 118E. 0F00.003C.0180.0800.0000.1066  ;2570  d_0    ; Prepare to set starting address to Zero
U 1066. 0000.003D.0180.0800.0000.11A0  ;2571 =0 call[set.start.addr] ; Go do it
          ;2572  j/find.mem.lp1.  ; Check to see if we are done
U 1067. 0003.003C.0180.09E8.0000.1198  ;2573   rc[0d]_0 ; ....
          ;2574 =0
          ;2575 NOT.FIRST.MS780E:
U 1068. 0000.003D.0180.0800.0000.11A0  ;2576  call[set.start.addr] ; Go set starting address to maximum
U 1069. 0000.003C.F9F0.2C00.0000.118F  ;2577  q_id[3e]  ; Get the number of MS780-Es found
U 118F. 0819.2014.0580.0800.0000.1190  ;2578  d_q+k[.1]  ; Increment this counter
U 1190. 0000.003C.F980.3C00.0000.1191  ;2579  id[3e]_d  ; Save the counter
U 1191. 0018.0038.11C0.0800.0000.1192  ;2580  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2581     ; ...were the I/O base addresses will be stored
U 1192. 081D.2000.7980.0800.0000.1193  ;2582  d_q-d,k[.30]  ; ... adjust pointer
U 1193. 0819.0014.7980.0800.0000.1194  ;2583  d_d+k[.30]  ; Point the SC to the ID register
U 1194. 0000.003C.F580.0800.0000.1195  ;2584  k[.a]   ; ...to receive I/O base address
U 1195. 0019.0014.F580.0800.0082.1196  ;2585  alu_d+k[.a],sc_alu ; ...
U 1196. 0810.0038.0180.0970.0000.1197  ;2586  d_rc[0e]  ; Get base address to d
U 1197. 0000.003C.0180.3400.0000.1198  ;2587  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2588
          ;2589 ;
          ;2590 ; Try next tr
          ;2591 ;
          ;2592 FIND.MEM.LP1:
U 1198. 0818.0014.0580.0A80.0000.1199  ;2593  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2594  alu_d.and.k[.f],
U 1199. 0019.0034.6180.0800.0010.119A  ;2595       clk.ubcc  ; Check if all done
U 119A. 0000.013C.0180.0800.0000.106A  ;2596  z?    ; Branch if yes
U 106A. 0000.003C.0180.0800.0000.1056  ;2597 =0 j/find.mem.loop  ; Try next TR
U 106B. 0000.003C.0180.0800.0000.1070  ;2598  nop
U 1070. 0000.003D.0180.0800.0000.103E  ;2599 =0 CALL[SBI.INIT]  ; Clear the SBI
U 1071. 0810.0038.0180.0908.0010.119B  ;2600  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 119B. 0000.013C.0180.0800.0000.1072  ;2601  z?    ; Check condition codes
U 1072. 0001.003E.0180.09F0.0000.0002  ;2602 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1073. 0000.003E.0180.0800.0000.0001  ;2603  return1    ; No MS780E found
          ;2604
```

```
            ;2605  .PAGE
            ;2606  .TOC "   Generate IO Address"
            ;2607 ;++
            ;2608 ; FUNCTIONAL DESCRIPTION:
            ;2609 ;
            ;2610 ; This routine is used to generate an SBI Configuration Register
            ;2611 ; address from a TR level.
            ;2612 ;
            ;2613 ; CALLING CONSTRAINT:
            ;2614 ;
            ;2615 ; =0
            ;2616 ;
            ;2617 ; INPUTS:
            ;2618 ;
            ;2619 ; D - Contains TR level
            ;2620 ;
            ;2621 ; OUTPUTS:
            ;2622 ;
            ;2623 ; D - Contains SBI IO address
            ;2624 ;--
            ;2625 GENERATE.IO:
U 119C, 0000.003C.89F8.0800.0084.719D  ;2626  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 119D, 0D00.003C.8D80.0800.0084.719E  ;2627  d_dal.sc,sc_k[.1f] ; Put TR level in place, setup to
U 119E, 0000.003C.0980.0800.0084.B19F  ;2628  sc_sc-k[.2]  ; insert bit 29
U 119F, 0801.001E.0180.0800.0000.0001  ;2629  d_d.ornot.mask,return1 ; and return
            ;2630
            ;2631
```

```
            ;2632  .PAGE
            ;2633  .TOC "   Set Starting Address"
            ;2634 ;**
            ;2635 ; FUNCTIONAL DESCRIPTION:
            ;2636 ;
            ;2637 ; This routine is used to set the starting address of the
            ;2638 ; memory to the specified value.
            ;2639 ;
            ;2640 ; CALLING CONSTRAINT:
            ;2641 ;
            ;2642 ; =0
            ;2643 ;
            ;2644 ; INPUTS:
            ;2645 ;
            ;2646 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2647 ;     (B Register Image divided by 2**16)
            ;2648 ; rc[0] - Contains Address of Register A
            ;2649 ;
            ;2650 ; OUTPUTS:
            ;2651 ;
            ;2652 ; d - Contains aligned starting address (B Register Image)
            ;2653 ;
            ;2654 ; SIDE EFFECTS:
            ;2655 ;
            ;2656 ; d,q,va, and sc are destroyed
            ;2657 ;--
            ;2658 SET.START.ADDR:
U 11A0, 0010,0038,6580,0970,0284,71A1  ;2659  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 11A1, 0000,003C,01F8,0803,0000,11A2  ;2660  va_va+4,q_0  ; Set address to Register B
            ;2661  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 11A2, 0D00,003C,0980,0800,0084,B1A3  ;2662       sc_sc-k[.2]  ; Setup to insert bit 14
U 11A3, 0801,001C,0180,0800,0000,11A4  ;2663  d_d.ornot.mask  ; Insert Write Enable bit
U 11A4, 0000,003E,0180,A800,0000,0001  ;2664  cache.p_d[long],return1 ; Set the starting address and return
            ;2665
```

```
            ;2666  .PAGE
            ;2667  .TOC "   ENABLE NEXT MS780-E"
            ;2668 ;++
            ;2669 ; FUNCTIONAL DESCRIPTION:
            ;2670 ;
            ;2671 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2672 ;     They will be tested  separately, according to the TR level at which
            ;2673 ;     they are located,  starting with the one  at the lowest level first.
            ;2674 ;          When  a test has finished  with the first unit, it will have to call
            ;2675 ;     this specific  routine  to see  if any other MS780-E unit is present
            ;2676 ;     on the SBI. If more units are  present, the first one  will be  dis-
            ;2677 ;     abled  by  setting  its  starting  address to maximum, the next unit
            ;2678 ;     will be  enabled  by  setting its starting address to 0 and the test
            ;2679 ;     will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2680 ;     for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2681 ;     ID registers 3A through 3D and  a count of the Total number of units
            ;2682 ;     found - 1 in register 3E. In this subroutine the SC register is used
            ;2683 ;     as  a  pointer  to  ID  registers 3A through 3D.
            ;2684 ;
            ;2685 ; CALLING CONSTRAINT:
            ;2686 ;
            ;2687 ;   =00
            ;2688 ;
            ;2689 ;--
            ;2690 ENABLE.NEXT.MS780E:
U 11A5, 0000,003C,F9F0,2C00,0000,11A6  ;2691  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2692  ALU_Q,    ; Check to see if it is zero
U 11A6, 0001,203C,0180,0800,0010,11A7  ;2693      CLK.UBCC  ; Check Condition Codes
U 11A7, 0000,013C,0180,0800,0000,1074  ;2694  Z?    ; ...for zero
U 1074, 0000,003C,0180,0800,0000,11A8  ;2695 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1075, 0000,003E,0180,0800,0000,0002  ;2696  RETURN2    ; Zero No more units to test
            ;2697 ENABLE.NEXT:
U 11A8, 0818,0038,C180,0800,0000,1076  ;2698  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1076, 0000,003D,0180,0800,0000,11A0  ;2699 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1077, 0818,0038,7980,0800,0000,11A9  ;2700  D_K[.30]  ; Prepare to point to the ID register holding
            ;2701      ; ...the I/O Base address of the next MS780-E
U 11A9, 0819,0014,1180,0800,0000,11AA  ;2702  D_D+K[.4]  ; ...
U 11AA, 0000,003C,F580,0800,0000,11AB  ;2703  K[.A]    ; ...
U 11AB, 0819,0014,F580,0800,0000,11AC  ;2704  D_D+K[.A]  ; ...
U 11AC, 0000,003C,F9F0,2C00,0000,11AD  ;2705  Q_ID[3E]  ; Get MS780-E Counter
            ;2706  D_D-Q,    ; D now holds the pointer to the ID register
U 11AD, 081D,0000,0180,0800,0082,11AE  ;2707   SC_ALU  ; ...
U 11AE, 0000,003C,01F0,2400,0000,11AF  ;2708  Q_ID(SC)  ; Q gets adrress of next MS780E
U 11AF, 0001,203C,0180,09F0,0000,11B0  ;2709  RC[0E]_Q  ; Save it also in RC[0E]
U 11B0, 0F03,003C,0180,09E8,0000,1078  ;2710  RC[0D]_0,D_0  ; Set starting address to zero
U 1078, 0000,003D,0180,0800,0000,11A0  ;2711 =0 CALL[SET.START.ADDR] ; ....
U 1079, 0F00,003C,F9F0,2C00,0000,11B1  ;2712  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 11B1, 081D,2008,0180,0800,0000,11B2  ;2713  D_Q-D-1    ; Subtract 1 from total
U 11B2, 0000,003C,F980,3C00,0000,107A  ;2714  ID[3E]_D  ; Adjust the pointer
U 107A, 0000,003D,0180,0800,0000,1133  ;2715 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 107B, 0000,003E,0180,0800,0000,0001  ;2716  RETURN1    ; Tell caller another unit was found
            ;2717
```

```
            ;2718  .PAGE
            ;2719  .TOC "   Select Controller"
            ;2720 ;**
            ;2721 ; FUNCTIONAL DESCRIPTION:
            ;2722 ;
            ;2723 ; This routine is used to put the memory system into the
            ;2724 ; specified configuration with respect to controller select.
            ;2725 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2726 ; a RETURN2 is made.
            ;2727 ;
            ;2728 ; CALLING CONSTRAINT:
            ;2729 ;
            ;2730 ; =00
            ;2731 ;
            ;2732 ; INPUTS:
            ;2733 ;
            ;2734 ; d - Contains value to write to bits<2:0> of Register A
            ;2735 ; rc[0e] - Address of Register A
            ;2736 ;
            ;2737 ; SIDE EFFECTS:
            ;2738 ;
            ;2739 ; sc,d,va are destroyed
            ;2740 ;--
            ;2741 SELECT.CNTRLR:
U 11B3, 0010.003B.0180.0970.0284,71B4  ;2742  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 11B4, 0801.001C.0180.0800.0000,11B5  ;2743  d_d.ornot.mask  ; Insert the enable bit into D reg
U 11B5, 0000.003C.0180.A800.0000,11B6  ;2744  cache.p_d[long]  ; Write the configuration Register
U 11B6, 0818.003B.5D80.0800.0000,11B7  ;2745  d_k[.7]   ; Get Misconfiguration bits
U 11B7, 0000.003C.61F8.0800.0084,71B8  ;2746  sc_k[.F],q_0  ; ...
U 11B8, 0D00.003C.0180.0800.0000,11B9  ;2747  d_dal.sc  ; ... D = x38000
U 11B9, 0000.003C.01E0.0800.0000,11BA  ;2748  q_d   ; Save mask
U 11BA, 0000.003C.0180.C800.0000,11BB  ;2749  d[long]_cache.p  ; Read CNFG A Reg and
            ;2750  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 11BB, 001D.2034.0180.0800.0010,11BC  ;2751   clk.ubcc ; ...
U 11BC, 0000.013C.0180.0800.0000,107C  ;2752  z?   ; Branch if no
U 107C, 0000.003E.0180.0800.0000,0001  ;2753 =0 RETURN1   ; Bad configuration
U 107D, 0000.003E.0180.0800.0000,0002  ;2754  RETURN2   ; Configuration ok
            ;2755
            ;2756
```

```
;2757  .PAGE
;2758  .TOC "    SELECT LOWER CONTROLLER"
;2759 ;*************************************************************
;2760 ;++
;2761 ;
;2762 ; Functional Description:
;2763 ; ----------------------
;2764 ;
;2765 ; This subroutine can be called to select the lower
;2766 ; Controller on a MS780-E/H.
;2767 ; If the Lower controller is misconfigured then a
;2768 ; RETURN1 will be made.  Otherwise a RETURN2
;2769 ;
;2770 ; Calling Constraint: =00
;2771 ; ------------------
;2772 ;
;2773 ;
;2774 ; Inputs:
;2775 ; ------
;2776 ;
;2777 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2778 ;
;2779 ; Outputs:
;2780 ; -------
;2781 ;
;2782 ; RC[0D] will have the address of CNFG Register C
;2783 ;        ( 2000x008 )
;2784 ;
;2785 ; Side Effects:
;2786 ; ------------
;2787 ;
;2788 ; Contents of the following registers will lost:
;2789 ;
;2790 ; D
;2791 ;
;2792 ; The Lower controller on the MS780-E/H will be configured
;2793 ; when the subroutine is exitted with a RETURN2
;2794 ;--
;2795 ;*************************************************************
;2796 =00
;2797 SELECT.LOWER:
;2798  D_K[ZERO],   ; Interleave mode value for CNFG Reg A
U 1028, 0818,0039,1980,0800,0000,11B3  ;2799   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1029, 0000,003E,0180,0800,0000,0001  ;2800  RETURN1    ; Lower Controller Misconfigured
U 102A, 0810,0038,0180,0970,0000,102B  ;2801  D_RC[0E]   ; Get MS780-E/H I/O Base address
;2802  RC[0D]_D+K[.8],   ; Set the address of CNFG Reg D
U 102B, 0019,0016,0180,09E8,0000,0002  ;2803   RETURN2   ; Let caller know that the controller
;2804       ; ...was configured properly
;2805
```

```
;2806  .PAGE
;2807  .TOC "   SELECT UPPER CONTROLLER"
;2808 ;*************************************************************
;2809 ;++
;2810 ;
;2811 ; Functional Description:
;2812 ; ----------------------
;2813 ;
;2814 ; This subroutine can be called to select the upper
;2815 ; Controller on a MS780-E/H.
;2816 ; If the Upper controller is misconfigured then a
;2817 ; RETURN1 will be made.  Otherwise a RETURN2
;2818 ;
;2819 ; Calling Constraint: =00
;2820 ; ------------------
;2821 ;
;2822 ;
;2823 ; Inputs:
;2824 ; ------
;2825 ;
;2826 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;2827 ;
;2828 ; Outputs:
;2829 ; -------
;2830 ;
;2831 ; RC[0D] will have the address of CNFG Register D
;2832 ;        ( 2000x010 )
;2833 ;
;2834 ; Side Effects:
;2835 ; ------------
;2836 ;
;2837 ; Contents of the following registers will lost:
;2838 ;
;2839 ; D
;2840 ;
;2841 ; The Upper controller on the MS780-E/H will be configured
;2842 ; when the subroutine is exitted with a RETURN2
;2843 ;--
;2844 ;*************************************************************
;2845 =00
;2846 SELECT.UPPER:
;2847  D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1030, 0818,0039,0980,0800,0000,11B3  ;2848   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1031, 0000,003E,0180,0800,0000,0001  ;2849  RETURN1     ; Upper Controller Misconfigured
U 1032, 0810,0038,0180,0970,0000,1033  ;2850  D_RC[0E]    ; Get MS780-E/H I/O Base address
;2851  RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 1033, 0019,0016,8580,09E8,0000,0002  ;2852   RETURN2    ; Let caller know that the controller
;2853       ; ...was configured properly
;2854
;2855 ;+
;2856 ; THIS ROUTINE LOOKS FOR A DEVICE ON THE SBI AND RETURNS WITH THE DEVICES
;2857 ; BASE ADDRESS IN R[0] IF IT FINDS ONE, OTHERWISE R[0] IS CLEAR.
;2858 ;-
;2859
;2860 LOOKFORDEVICE:
```

```
U 11BD. 0818.0038.7580.0800.0000.11BE  ;2861  D_K[.20]
U 11BE. 0B00.003C.0180.0800.0000.11BF  ;2862  D_D.SWAP    ; GENERATE HIGH WORD OF DEVICE ADDRESS
U 11BF. 0001.003C.0180.0A80.0000.11C0  ;2863  R[0]_D      ; SAVE
U 11C0. 0818.0038.75F8.0800.0000.11C1  ;2864  D_K[.20],Q_0    ; GENERATE INITIAL LOW WORD
U 11C1. 0000.003C.0180.0800.0084.71C2  ;2865  SC_K[.8]    ; ...
U 11C2. 0D00.003C.0180.0800.0000.11C3  ;2866  D_DAL.SC    ; ...
U 11C3. 081C.2030.0180.0A00.0000.11C4  ;2867  D_D.OR.R[0]    ; D= 20002000
U 11C4. 0001.003C.0180.0A80.0000.11C5  ;2868  R[0]_D     ; SAVE (ADRS OF DEVICE AT TR=1)
U 11C5. 0818.0038.7580.0800.0000.11C6  ;2869  D_K[.20]    ; GENERATE INCREMENT FOR DEVICE ADDRESS
U 11C6. 0D00.003C.0180.0800.0900.11C7  ;2870  D_DAL.SC    ; D= 2000
U 11C7. 0000.003C.01E0.0800.0000.11C8  ;2871  Q_D     ; SAVE INCREMENT IN Q
U 11C8. 0000.003C.0180.0A00.0200.11C9  ;2872  VA_ALU,ALU_R[0]    ; LOAD THE DEVICE ADDRESS
U 11C9. 0018.0038.6180.0A90.0000.11CA  ;2873  R[2]_K[.F]    ; SET THE LOOP COUNT
U 11CA. 0000.003C.8980.0800.0084.71CB  ;2874  SC_K[.D]
           ;2875 TRYAGAIN:
U 11CB. 0000.003C.0180.0A00.0200.11CC  ;2876  VA_ALU,ALU_R[0]    ; LOAD THE ADDRESS
U 11CC. 0003.001C.0180.0AF8.0000.11CD  ;2877  R[0F]_NOT.MASK    ; SET THE TIMEOUT TRAP EXPECTED FLAG
U 11CD. 0000.003C.0180.C800.0000.11CE  ;2878  D[LONG]_CACHE.P    ; READ THE ADDRESS
U 11CE. 0000.003C.0180.0A78.0010.11CF  ;2879  ALU_R[0F],CLK.UBCC   ; WAS THERE A TIMEOUT?
U 11CF. 0000.013C.0180.0800.0000.107E  ;2880  Z?     ; BRANCH IF YES
U 107E. 0000.003C.0180.0800.0000.11DC  ;2881 =0 J/FOUNDONE    ; EXIT
           ;2882 FOUNDMEMORY:
U 107F. 0818.0038.4580.0800.0000.11D0  ;2883  D_K[.8000]    ; CLEAR THE TIMEOUT BIT
U 11D0. 0200.003C.0180.0800.0000.11D1  ;2884  D_D.RIGHT2
U 11D1. 0600.003C.0180.0800.0000.11D2  ;2885  D_D.RIGHT
U 11D2. 0000.003C.6580.3C00.0000.11D3  ;2886  ID[SBI.ERR]_D    ; ...
U 11D3. 0800.003C.0180.0A00.0000.11D4  ;2887  D_R[0]     ; GET THE CURRENT ADDRESS
U 11D4. 001D.0014.0180.0A80.0200.11D5  ;2888  ALU_D+Q,VA_ALU,R[0]_ALU   ; INCREMENT THE ADDRESS
U 11D5. 0800.003C.0180.0A10.0000.11D6  ;2889  D_R[2]     ; GET THE LOOP COUNT
U 11D6. 0019.0000.0580.0A90.0010.11D7  ;2890  R[2]_D-K[.1],CLK.UBCC   ; DECREMENT AND CHECK
U 11D7. 0000.013C.0180.0800.0000.1080  ;2891  Z?     ; BRANCH IF DONE
U 1080. 0000.003C.0180.0800.0000.11CB  ;2892 =0 J/TRYAGAIN    ; CONTINUE LOOKING FOR DEVICE
U 1081. 0003.003C.0180.0A80.0084.71D8  ;2893  R[0]_0,SC_K[.8]    ; NO DEVICE AVAILABLE
           ;2894 CLEARERROR:
U 11D8. 0818.0038.65F8.0800.0000.11D9  ;2895  D_K[.10],Q_0    ; SETUP TO CLEAR SBI ERROR REG
U 11D9. 0D00.003C.0180.0800.0000.11DA  ;2896  D_DAL.SC
U 11DA. 0819.0030.3180.0800.0000.11DB  ;2897  D_D.OR.K[.40]
U 11DB. 0000.003E.6580.3C00.0000.0001  ;2898  ID[SBI.ERR]_D,RETURN1   ; CLEAR ERROR REG AND RETURN
           ;2899 FOUNDONE:
U 11DC. 0200.003C.0180.0800.0000.11DD  ;2900  D_D.RIGHT2    ; CHECK BIT 5 TO SEE IF ITS MEMORY
U 11DD. 0200.003C.0180.0800.0000.11DE  ;2901  D_D.RIGHT2    ; ...
U 11DE. 0819.0034.5D80.0800.0000.11DF  ;2902  D_D.AND.K[.7]    ; ...
U 11DF. 0000.193C.0180.0800.0000.1018  ;2903  D3-0?
U 1018. 0000.003C.0180.0800.0000.107F  ;2904 =1000 J/FOUNDMEMORY    ; MS780 OR MA780
U 1019. 0000.003C.0180.0800.0000.107F  ;2905  J/FOUNDMEMORY    ; MS780 OR MA780
U 101A. 0000.003C.0180.0800.0000.11E0  ;2906  J/UBAMBA
U 101B. 0000.003C.0180.0800.0000.107F  ;2907  J/FOUNDMEMORY
U 101C. 0000.003C.0180.0800.0000.107F  ;2908  J/FOUNDMEMORY
U 101D. 0000.003C.0180.0800.0000.107F  ;2909  J/FOUNDMEMORY
U 101E. 0000.003C.0180.0800.0000.107F  ;2910  J/FOUNDMEMORY
U 101F. 0000.003C.0180.0800.0000.107F  ;2911  J/FOUNDMEMORY
           ;2912 UBAMBA:
U 11E0. 0000.003C.0180.0800.0084.71D8  ;2913  SC_K[.8],J/CLEARERROR
           ;2914
           ;2915
```

```
            ;2916 ;*
            ;2917 ; THIS ROUTINE IS USED TO WAIT FOR THE UNIBUS TO POWER BACK UP AFTER AN
            ;2918 ; UNJAM SEQUENCE
            ;2919 ;-
            ;2920
            ;2921 WAITFORUNIBUS:
U 11E1, 0818,0038,C180,0800,0000,11E2  ;2922  D_K[.FFFF]
U 11E2, 0500,003C,0180,0800,0000,11E3  ;2923  D_D.LEFT
U 11E3, 0819,0000,0580,0800,0010,11E4  ;2924 LOOPUN: D_D-K[.1],CLK.UBCC
U 11E4, 0000,013C,0180,0800,0000,1082  ;2925  Z?
U 1082, 0000,003C,0180,0800,0000,11E3  ;2926 =0 J/LOOPUN
U 1083, 0000,003E,0180,0800,0000,0001  ;2927  RETURN1
            ;2928
            ;2929
            ;2930 ;*
            ;2931 ; THE FOLLOWING ROUTINE CALLS THE CONSOLE TO TYPE A MESSAGE ON THE
            ;2932 ; TERMINAL. IT IS ENTERED WITH THE D REGISTER CONTAINING THE MESSAGE
            ;2933 ; NUMBER.
            ;2934 ;-
            ;2935
U 11E5, 0000,003C,F580,0800,0084,71E6  ;2936 MSGCOM: SC_K[.A]
            ;2937 MSGCOM_1:
U 11E6, 0000,003C,0180,0800,0100,11E7  ;2938  FE_SC
U 11E7, 0000,003C,D9F8,0800,0084,71E8  ;2939  SC_K[.9],Q_0
U 11E8, 0D00,003C,0180,0800,0000,11E9  ;2940  D_DAL.SC   ; PUT MSG NUMBER IN BYTE 1
U 11E9, 0000,003C,C1F0,2C00,0000,11EA  ;2941  Q_ID[T0]   ; TYPEOUT FLAG SET?
U 11EA, 0C01,203C,01E0,0800,0010,11EB  ;2942  ALU_Q,CLK.UBCC,Q_D,D_Q
U 11EB, 0819,1B14,0580,0800,0000,1027  ;2943  ALU.N?,D_D+K[.1]
U 1027, 0C00,003E,0180,0800,0000,0001  ;2944 =0111 RETURN1,D_Q   ; NO TYPEOUT
U 102F, 0C00,003C,C180,3C00,0081,11EC  ;2945  ID[T0]_D,D_Q,SC_FE
U 11EC, 0819,0030,1D80,0800,0000,105E  ;2946  D_D.OR.K[SC],J/CALLCON  ; GO TYPE THE MESSAGE
            ;2947 MSGCOM_CRLF:
U 11ED, 0018,0038,65C0,0800,0000,11EE  ;2948  Q_K[.10]
U 11EE, 0019,2030,0980,0800,0082,11E6  ;2949  SC_Q.OR.K[.2],J/MSGCOM_1
U 11EF, 0000,003C,6580,0800,0084,71E6  ;2950 RCECOM: SC_K[.10],J/MSGCOM_1
            ;2951 RCECOM_CRLF:
U 11F0, 0000,003C,2180,0800,0084,71E6  ;2952  SC_K[.14],J/MSGCOM_1
            ;2953
            ;2954 ;**
            ;2955 ; THIS ROUTINE IS USED TO DECREMENT THE TYPEOUT FLAG
            ;2956 ;--
U 11F1, 0000,003C,C1F0,2C00,0000,11F2  ;2957 DECFLG: Q_ID[T0]   ; READ THE FLAG
U 11F2, 0819,2000,0580,0800,0000,11F3  ;2958  D_Q-K[.1]   ; DECREMENT IT
U 11F3, 0000,003E,C180,3C00,0000,0001  ;2959  ID[T0]_D,RETURN1  ; RESTORE IT AND RETURN
            ;2960
```

```
            ;2961  .PAGE "TEST 20B SBI INVALIDATE"
            ;2962 ;*****************************************************************
            ;2963 ;++
            ;2964 ;
            ;2965 ; SBI INVALIDATE
            ;2966 ;
            ;2967 ;TEST DESCRIPTION
            ;2968 ; THIS TEST CHECKS THAT THE CACHE IS INVALIDATED WHEN A
            ;2969 ; NEXUS ON THE SBI OTHER THAN THE CPU DOES A WRITE
            ;2970 ; FUNCTION TO MEMORY WHICH IS A HIT IN THE CACHE.
            ;2971 ; LOCATION ZERO IS VALIDATED. THE UBA THEN DOES A WRITE TO
            ;2972 ; MEMORY LOCATION ZERO. THE CPU READS THE LOCATION.
            ;2973 ; THE MAINTENANCE REGISTER HIT/MISS BITS ARE THEN
            ;2974 ; CHECKED TO SEE IF THE CACHE WAS INVALIDATED.
            ;2975 ;
            ;2976 ;LOGIC DESCRIPTION
            ;2977 ; THE WRITE INVALID LOGIC ON SBHA, SBHM, SBLB, SBLJ AND
            ;2978 ; ASSOCIATED LOGIC IS CHECKED.
            ;2979 ;
            ;2980 ;ERROR DESCRIPTION
            ;2981 ; EXPECTED MAINTENANCE REGISTER DATA
            ;2982 ; RECEIVED MAINTENANCE REGISTER DATA
            ;2983 ;
            ;2984 ;--
            ;2985 ;*****************************************************************
            ;2986
            ;2987 =0
U 1084, 0018,0039,0980,09F8,0000,10EA  ;2988 INVAL: NEWTST[.2]
U 1085, 0000,003C,0180,0800,0000,1086  ;2989  NOP
U 1086, 0000,003D,0180,0800,0000,1046  ;2990 =0 CALL,J/MAINTREGSETUP
U 1087, 0000,003C,0180,0800,0000,1088  ;2991  NOP
U 1088, 0000,003D,0180,0800,0000,11BD  ;2992 =0 CALL,J/LOOKFORDEVICE  ; TRY AND FIND A UBA
            ;2993 VCHECKR0:
U 1089, 0000,003C,0180,0A00,0210,11F4  ;2994  VA_ALU,ALU_R[0],CLK.UBCC ; WAS THERE A DEVICE?
U 11F4, 0000,013C,0180,0800,0000,108A  ;2995  Z?    ; BRANCH IF NO
U 108A, 0000,003C,0180,0800,0000,11F5  ;2996 =0 J/VCHECKCODE1   ; CHECK IF ITS A UBA
U 108B, 0003,003C,0180,09F0,0000,109A  ;2997  RC[0E]_0,J/VEND003  ; NO UBA AVAILABLE
            ;2998
            ;2999 VCHECKCODE1:
U 11F5, 0000,003C,0180,C800,0000,11F6  ;3000  D[LONG]_CACHE.P   ; READ THE CONFIG REG
U 11F6, 0200,003C,0180,0800,0000,11F7  ;3001  D_D.RIGHT2   ; CHECK BITS 3 AND 4
U 11F7, 0600,003C,0180,0800,0000,11F8  ;3002  D_D.RIGHT   ; ...
U 11F8, 0000,193C,0180,0800,0000,103C  ;3003  D1-0?    ; UBA = 01
U 103C, 0000,003C,0180,0800,0000,1201  ;3004 =1100 J/VLOOKAGAIN   ; THIS IS MBA
U 103D, 0000,003C,0180,C800,0000,11F9  ;3005  D[LONG]_CACHE.P   ; GET CONFIG REG AGAIN
U 11F9, 0819,0034,0D80,0800,0000,11FA  ;3006 = D_D.AND.K[.3]   ; GENERATE THE BASE ADDRESS
U 11FA, 0819,0030,1180,0800,0000,11FB  ;3007  D_D.OR.K[.4]   ; OF UNIBUS SPACE
U 11FB, 0000,003C,21F8,0800,0084,71FC  ;3008  SC_K[.14],Q_0
U 11FC, 0000,003C,0980,0800,0084,B1FD  ;3009  SC_SC-K[.2]
U 11FD, 0D00,003C,D980,0800,0000,11FE  ;3010  D_DAL.SC,K[.9]
U 11FE, 0000,003C,D980,0800,0084,91FF  ;3011  SC_SC+K[.9]
U 11FF, 0000,003C,0980,0800,0084,9200  ;3012  SC_SC+K[.2]
U 1200, 0001,001C,0180,0980,0000,108E  ;3013  RC[0]_ALU,ALU_D.ORNOT.MASK,J/VSTART ; ...
            ;3014
            ;3015 VLOOKAGAIN:
```

```
U 1201, 0000,003C,0180,0800,0084,7202  ;3016  SC_K[.8]
U 1202, 0818,0038,75F8,0800,0000,1203  ;3017  D_K[.20],Q_0
U 1203, 0D00,003C,0180,0800,0000,108C  ;3018  D_DAL.SC
U 108C, 0000,003D,89E0,0800,0084,707F  ;3019 =0 Q_D,SC_K[.D],CALL,J/FOUNDMEMORY
U 108D, 0000,003C,0180,0800,0000,1089  ;3020  J/VCHECKR0
          :3021
          :3022 =0
          :3023 VSTART:
U 108E, 0000,003D,0180,0800,0000,11E1  ;3024  CALL,J/WAITFORUNIBUS
U 108F, 0000,003C,75F0,2C00,0000,1204  ;3025  Q_ID[MAINT]   ; SET THE SBI INVALIDATE ENABLE BIT
U 1204, 0000,003C,2180,0800,0084,7205  ;3026  SC_K[.14]
U 1205, 0000,003C,0580,0800,0084,9206  ;3027  SC_SC+K[.1]
U 1206, 0801,201C,0180,0800,0000,1207  ;3028  D_Q.ORNOT.MASK
U 1207, 0000,003C,7580,3C00,0000,1208  ;3029  ID[MAINT]_D   ; ...
U 1208, 0800,003C,F580,0A00,0084,7209  ;3030  D_R[0],SC_K[.A]   ; GENERATE ADDRESS OF THE UBA
U 1209, 0000,003C,0580,0800,0084,920A  ;3031  SC_SC+K[.1]
U 120A, 0001,001C,0180,0980,0000,120B  ;3032  RC[1]_D.ORNOT.MASK  ; SAVE IN RC[1] = 200XX800
U 120B, 0800,003C,0180,0A00,0000,120C  ;3033  D_R[0]    ; GET BASE ADDRESS AGAIN
U 120C, 0819,0030,3180,0800,0000,120D  ;3034  D_D.OR.K[.40]   ; GENERATE ADDRESS OF BUFFERED DATA
U 120D, 0019,0030,1180,0A98,0000,120E  ;3035  R[3]_ALU,ALU_D.OR.K[.4]  ; PATH REGISTER 1
U 120E, 0818,0038,4580,0800,0000,120F  ;3036  D_K[.8000]   ; GENERATE BASE CONTENTS OF MAP
U 120F, 0819,0030,7580,0800,0000,1210  ;3037  D_D.OR.K[.20]
U 1210, 0000,003C,65F8,0800,0084,7211  ;3038  SC_K[.10],Q_0
U 1211, 0D00,003C,0180,0800,0000,1212  ;3039  D_DAL.SC
U 1212, 0001,003C,0180,0AA8,0000,1090  ;3040  R[5]_D    ; SAVE IN R5 = 80200000
U 1090, 0000,003D,0180,0800,0000,10FF  ;3041 =0 ERLOOP
U 1091, 0000,003C,75F0,2C00,0000,1213  ;3042  Q_ID[MAINT]
U 1213, 0C00,003C,0180,0800,0000,1214  ;3043  D_Q
U 1214, 0000,003C,2180,0800,0084,7215  ;3044  SC_K[.14]
U 1215, 0000,003C,0580,0800,0084,9216  ;3045  SC_SC+K[.1]
U 1216, 0801,001C,0180,0800,0000,1217  ;3046  D_D.ORNOT.MASK
U 1217, 0000,003C,7580,3C00,0000,1218  ;3047  ID[MAINT]_D   ;ENABLE SBI INVALIDATES
U 1218, 0001,003C,0180,09F0,0000,1219  ;3048  RC[0E]_D   ;SAVE EXPECTED DATA
U 1219, 0018,0038,1980,0800,0200,121A  ;3049  VA_K[ZERO]
U 121A, 0000,003C,0180,9800,0000,121B  ;3050  MCT/VALIDATE   ;VALIDATE LOCATION ZERO
U 121B, 0000,003C,0180,0803,0000,121C  ;3051  VA_VA+4
U 121C, 0000,003C,0180,9800,0000,121D  ;3052  MCT/VALIDATE   ;AND NEXT LOCATION
U 121D, 0010,0038,0180,0908,0200,121E  ;3053  VA_RC[1]   ;LOAD MAP REGISTER ADDRESS
U 121E, 0800,003C,0180,0A28,0000,121F  ;3054  D_R[5]    ;DATA FOR MAP REGISTER
U 121F, 0000,003C,0180,A800,0000,1221  ;3055  MCT/WRITE.P   ;LOAD MAP REGISTER
U 1221, 0810,0038,0180,0900,0200,1222  ;3056  VA_RC[0],D_ALU   ;LOAD UNIBUS ADDRESS
U 1222, 0000,403C,0180,A800,0000,1223  ;3057  CACHE.P_D[WORD]   ;LOAD BUFFERED DATA PATH
U 1223, 0019,0014,0980,0800,0200,1225  ;3058  VA_D+K[.2]   ;WITH A QUADWORD
U 1225, 0000,403C,0180,A800,0000,1226  ;3059  CACHE.P_D[WORD]
U 1226, 0019,0014,1180,0800,0200,1227  ;3060  VA_D+K[.4]
U 1227, 0000,403C,0180,A800,0000,1229  ;3061  CACHE.P_D[WORD]
U 1229, 0019,0014,D580,0800,0200,122A  ;3062  VA_D+K[.6]
U 122A, 0000,403C,0180,A800,0000,122B  ;3063  CACHE.P_D[WORD]
U 122B, 0818,0038,F580,0800,0000,122D  ;3064  D_K[.A]
U 122D, 0819,0000,0580,0800,0010,122E  ;3065 WT: D_D-K[.1],CLK.UBCC  ;WAIT FOR WRITE TO COMPLETE
U 122E, 0000,013C,0180,0800,0000,1092  ;3066  Z?
U 1092, 0000,003C,0180,0800,0000,122D  ;3067 =0 J/WT
U 1093, 0018,0038,1980,0800,0200,122F  ;3068  VA_K[ZERO]
U 122F, 0000,003C,0180,C800,0000,1231  ;3069  MCT/READ.P   ;READ 0 TO CHECK FOR HIT
U 1231, 0000,003C,75F0,2C00,0000,1232  ;3070  Q_ID[MAINT]
```

```
U 1232, 0001,203C,0180,09E8,0000,1233  ;3071  RC[0D]_Q    ;SAVE RECEIVED DATA
U 1233, 0810,0038,0180,0970,0000,1235  ;3072  D_RC[0E]    ;GET EXPECTED DATA
U 1235, 001D,0000,0180,0800,0010,1236  ;3073  ALU_D-Q,CLK.UBCC
U 1236, 0000,013C,0180,0800,0000,1094  ;3074  Z?
U 1094, 0000,003D,0180,0800,0000,110B  ;3075 =0 ERROR2     ;DATA INCORRECT
U 1095, 0000,003C,0180,0800,0000,109D  ;3076  J/VEND005
          ;3077 =0
          ;3078 VEND004:
U 1096, 0000,003D,0180,0800,0000,11F1  ;3079  CALL[DECFLG]
U 1097, 0000,003C,0180,0800,0000,1098  ;3080  NOP
U 1098, 0818,0039,0580,0800,0000,11ED  ;3081 =0 TYP_MSG_1_CRLF    ;TYPE THE NO MEMORY MESSAGE
U 1099, 0000,003C,0180,0800,0000,109D  ;3082  J/VEND005
          ;3083 =0
U 109A, 0000,003D,0180,0800,0000,11F1  ;3084 VEND003:CALL[DECFLG]
U 109B, 0000,003C,0180,0800,0000,109C  ;3085  NOP
U 109C, 0818,0039,0980,0800,0000,11ED  ;3086 =0 TYP_MSG_2_CRLF    ;TYPE THE NO UBA MESSAGE
U 109D, 0000,003C,0180,0800,0000,109E  ;3087 VEND005: NOP
          ;3088
          ;3089
```

```
                ;3090 .PAGE "TEST 20C SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR0"
                ;3091 ;*****************************************************************
                ;3092 ;**
                ;3093 ;
                ;3094 ; SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR0
                ;3095 ;
                ;3096 ; TEST DESCRIPTION
                ;3097 ; THIS CHECKS THAT THE CACHE IS INVALIDATED WHEN A WRITE-MISS
                ;3098 ; PARITY ERROR IN GROUP 0 IS DETECTED. ADDRESS 1004 IS FIRST VALIDATED.
                ;3099 ; THEN THE REVERSE PARITY BITS ARE SET AND ADDRESS 4 IS WRITTEN WITH
                ;3100 ; A DIFFERENT DATA PATTERN THAN ADDRESS 1004 WAS VALIDATED WITH.
                ;3101 ; THE REVERSE PARITY IS THEN DISABLED AND ADDRESS 1004 IS READ TO SEE
                ;3102 ; THAT THE DATA CHANGED. LOCATION 1004 SHOULD STILL BE A HIT IN GROUP 0
                ;3103 ; SINCE THE REVERSE PARITY FUNCTION DISABLES THE WRITE PULSE TO THE
                ;3104 ; ADDRESS MATRIX BUT NOT THE DATA MATRIX.
                ;3105 ;
                ;3106 ;LOGIC DESCRIPTION
                ;3107 ; THE WRITE-MISS PARITY ERROR FF ON SBLR AND ASSOCIATED
                ;3108 ; LOGIC ON M8218 AND M8219 ARE TESTED.
                ;3109 ;
                ;3110 ; ERROR DESCRIPTION
                ;3111 ; DATA: EXPECTED READ DATA
                ;3112 ;       RECEIVED READ DATA
                ;3113 ;       MAINTENANCE REGISTER
                ;3114 ;--
                ;3115 ;*****************************************************************
                ;3116 =0
U 109E. 0018,0039,0D80,09F8,0000,10EA   ;3117 WRMSPE: NEWTST[.3]
U 109F. 0000,003C,0180,0800,0000,10A0   ;3118  NOP
U 10A0. 0000,003D,0180,0800,0000,1046   ;3119 =0 CALL,J/MAINTREGSETUP  ; SETUP SOME REGISTERS
U 10A1. 0F18,0038,1980,0800,0200,1237   ;3120  VA_K[ZERO],D_0
U 1237. 0000,003C,0180,9000,0000,1239   ;3121  MCT/INVALIDATE
U 1239. 0000,003C,8580,0800,0084,723A   ;3122  SC_K[.C]
U 123A. 0818,0038,1180,0800,0000,123B   ;3123  D_K[.4]
U 123B. 0801,001C,0180,0800,0000,123D   ;3124  D_D.ORNOT.MASK
U 123D. 0001,003C,0180,0980,0000,123E   ;3125  RC[0]_D    ; INIT THE ADDRESS FOR VALIDATION
U 123E. 0000,003C,6580,0800,0084,723F   ;3126  SC_K[.10]   ; INIT THE REVERSE PARITY BITS
U 123F. 0818,0038,01F8,0800,0000,1240   ;3127  D_K[.8],Q_0
U 1240. 0D00,003C,0180,0800,0000,1241   ;3128  D_DAL.SC
U 1241. 0001,003C,0180,0988,0000,1242   ;3129  RC[1]_D    ; SAVE
U 1242. 0018,0038,1980,09F0,0000,10A2   ;3130  RC[0E]_K[ZERO]   ; SET THE EXPECTED DATA
U 10A2. 0000,003D,0180,0800,0000,10FF   ;3131 =0 ERLOOP
U 10A3. 0010,0038,0180,0900,0200,1243   ;3132 VLOOP1: VA_RC[0]   ; LOAD ADDRESS TO VALIDATE
U 1243. 0818,0038,0580,0800,0000,1244   ;3133  D_K[.1]    ; GET DATA PATTERN TO WRITE
U 1244. 0000,003C,0180,9800,0000,1245   ;3134  MCT/VALIDATE   ; VALIDATE THE LOCATION
U 1245. 0810,0038,75F0,2D08,0000,1246   ;3135  Q_ID[MAINT],D_RC[01]  ; GET CURRENT VALUE OF MAINTENANCE REG
U 1246. 081D,0030,0180,0800,0000,1247   ;3136  D_D.OR.Q   ; COMBINE WITH REVERSE PARITY BITS
U 1247. 0000,003C,7580,3C00,0000,1248   ;3137  ID[MAINT]_D   ; SET REVERSE PARITY BITS
U 1248. 0F18,0038,1180,0800,0200,1249   ;3138  VA_K[.4],D_0   ; LOAD VA WITH ADDRESS 4
U 1249. 0000,003C,0180,A800,0000,124A   ;3139  CACHE.P_D[LONG]   ; WRITE LOCATION 4
U 124A. 0018,0038,5DC0,0800,0000,124B   ;3140  Q_K[.7]
U 124B. 0019,2000,05C0,0800,0010,124C   ;3141 NOPCNT: Q_Q-K[.1],CLK.UBCC  ; WAIT FOR WRITE TO COMPLETE
U 124C. 0000,013C,0180,0800,0000,10A4   ;3142  Z?
U 10A4. 0000,003C,0180,0800,0000,124B   ;3143 =0 J/NOPCNT
U 10A5. 0800,003C,0180,0A30,0000,124D   ;3144  D_R[6]    ; GET DATA TO CLEAR REVERSE PARITY
```

```
U 124D. 0010.0038.7580.3D00.0200.124E  ;3145  VA_RC[0],ID[MAINT]_D  ; LOAD ADDR AND CLEAR REV PARITY BITS
U 124E. 0000.003C.0180.0800.0000.124F  ;3146  NOP
U 124F. 0000.003C.0180.C800.0000.1250  ;3147  D[LONG]_CACHE.P    ; READ THE LOCATION
U 1250. 0000.003C.75F0.2C00.0000.1251  ;3148  Q_ID[MAINT]    ; READ THE MAINT REGISTER
U 1251. 0001.203C.0180.09E0.0000.1252  ;3149  RC[0C]_Q
U 1252. 0010.0038.01C0.0970.0000.1253  ;3150  Q_RC[0E]    ; GET EXPECTED DATA
U 1253. 001D.2000.0180.0800.0010.1254  ;3151  ALU_Q-D,CLK.UBCC   ; CHECK
U 1254. 0001.013C.0180.09E8.0000.10A6  ;3152  Z?,RC[0D]_D
U 10A6. 0000.003D.0180.0800.0000.1109  ;3153 =0 ERROR1     ; READ DATA INCORRECT
          ;3154 VEND001:
U 10A7. 0000.003C.0180.0800.0000.10A8  ;3155  NOP     ; DONE
          ;3156
          ;3157
          ;3158
```

```
                ;3159 .PAGE "TEST 20D SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR1"
                ;3160 ;*****************************************************************
                ;3161 ;**
                ;3162 ;
                ;3163 ; SBI INVALIDATE ON WRITE-MISS PARITY ERROR IN GR1
                ;3164 ;
                ;3165 ; TEST DESCRIPTION
                ;3166 ; THIS TEST CHECKS THAT THE CACHE IS INVALIDATED ON A WRITE-MISS
                ;3167 ; PARITY ERROR IN GROUP 1.
                ;3168 ; ADDRESS 1004 IS FIRST VALIDATED. THEN THE REVERSE PARITY
                ;3169 ; BITS ARE SET AND LOCATION 4 IS WRITTEN WITH A DIFFERENT DATA
                ;3170 ; PATTERN THAN ADDRESS 1004 WAS VALIDATED WITH. THE REVERSE PARITY
                ;3171 ; IS THEN DISABLED AND LOCATION 1004 IS READ TO SEE THAT THE
                ;3172 ; DATA CHANGED. LOCATION 1004 SHOULD STILL BE A HIT IN GROUP 1
                ;3173 ; SINCE THE REVERSE PARITY FUNCTION DISABLES THE WRITE PULSE
                ;3174 ; TO THE ADDRESS MATRIX BUT NOT THE DATA MATRIX.
                ;3175 ;
                ;3176 ;LOGIC DESCRIPTION
                ;3177 ; THE WRITE-MISS PARITY ERROR FF ON SBLR AND ASSOCIATED
                ;3178 ; LOGIC ON M8219 AND M8218 ARE TESTED.
                ;3179 ;
                ;3180 ; ERROR DESCRIPTION
                ;3181 ;
                ;3182 ; DATA: EXPECTED READ DATA
                ;3183 ;       RECEIVED READ DATA
                ;3184 ;       MAINTENANCE REGISTER
                ;3185 ;--
                ;3186 ;*****************************************************************
                ;3187 =0
U 10A8, 0018,0039,0D80,09F8,0000,10EA   ;3188 WMSPE1: NEWTST[.3]
U 10A9, 0000,003C,0180,0800,0000,10AA   ;3189  NOP
U 10AA, 0000,003D,0180,0800,0000,1046   ;3190 =0 CALL,J/MAINTREGSETUP  ; SETUP SOME REGISTERS
U 10AB, 0F18,0038,1980,0800,0200,1255   ;3191  VA_K[ZERO],D_0
U 1255, 0000,003C,0180,9000,0000,1256   ;3192  MCT/INVALIDATE
U 1256, 0818,0038,4580,0800,0000,1257   ;3193  D_K[.8000]   ; SETUP TO FORCE REPLACE G1 AND
U 1257, 0500,003C,8980,0800,0084,7258   ;3194  D_D.LEFT,SC_K[.D]  ; FORCE MISS GROUP 0
U 1258, 0801,001C,0180,0800,0000,1259   ;3195  D_D.ORNOT.MASK
U 1259, 0001,003C,7580,3EA0,0000,125A   ;3196  R[4]_D,ID[MAINT]_D  ; ...
U 125A, 0000,003C,8580,0800,0084,725B   ;3197  SC_K[.C]
U 125B, 0818,0038,1180,0800,0000,125C   ;3198  D_K[.4]
U 125C, 0801,001C,0180,0800,0000,125D   ;3199  D_D.ORNOT.MASK
U 125D, 0001,003C,0180,0980,0000,125E   ;3200  RC[0]_D    ; INIT THE ADDRESS FOR VALIDATION
U 125E, 0000,003C,6580,0800,0084,725F   ;3201  SC_K[.10]   ; INIT THE REVERSE PARITY IN
U 125F, 0818,0038,09F8,0800,0000,1260   ;3202  D_K[.2],Q_0   ; THE MAINTENANCE REGISTER
U 1260, 0D00,003C,0180,0800,0000,1261   ;3203  D_DAL.SC
U 1261, 0001,003C,0180,0988,0000,1262   ;3204  RC[1]_D    ; SAVE
U 1262, 0018,0038,1980,09F0,0000,10AC   ;3205  RC[0E]_K[ZERO]   ; GENERATE THE EXPECTED DATA
U 10AC, 0000,003D,0180,0800,0000,10FF   ;3206 =0 ERLOOP
U 10AD, 0010,0038,0180,0900,0200,1263   ;3207 VLOOP2: VA_RC[0]   ; LOAD ADDRESS TO VALIDATE
U 1263, 0818,0038,0580,0800,0000,1264   ;3208  D_K[.1]    ; GET DATA PATTERN TO WRITE
U 1264, 0000,003C,0180,9800,0000,1265   ;3209  MCT/VALIDATE   ; VALIDATE THE LOCATION
U 1265, 0810,0038,75F0,2D08,0000,1266   ;3210  Q_ID[MAINT],D_RC[01]  ; GET CURRENT VALUE OF MAINTENANCE REG
U 1266, 081D,0030,0180,0800,0000,1267   ;3211  D_D.OR.Q   ; COMBINE WITH REVERSE PARITY BITS
U 1267, 0000,003C,7580,3C00,0000,1268   ;3212  ID[MAINT]_D   ; SET REVERSE PARITY BITS
U 1268, 0F18,0038,1180,0800,0200,1269   ;3213  VA_K[.4],D_0   ; LOAD THE VA WITH ADDRESS 4
```

```
U 1269. 0000.003C.0180.A800.0000.126A  ;3214  CACHE.P_D[LONG]   ; WRITE TO LOCATION 4
U 126A. 0018.0038.5DC0.0800.0000.126B  ;3215  Q_K[.7]
U 126B. 0019.2000.05C0.0800.0010.126C  ;3216 NPCNT1: Q_Q-K[.1],CLK.UBCC  ; WAIT FOR THE WRITE TO COMPLETE
U 126C. 0000.013C.0180.0800.0000.10AE  ;3217  Z?
U 10AE. 0000.003C.0180.0800.0000.126B  ;3218 =0 J/NPCNT1
U 10AF. 0800.003C.0180.0A20.0000.126D  ;3219  D_R[4]    ; GET DATA TO CLEAR REVERSE PARITY
U 126D. 0010.0038.7580.3D00.0200.126E  ;3220  VA_RC[0],ID[MAINT]_D  ; LOAD ADDR AND CLEAR REV PARITY BITS
U 126E. 0000.003C.0180.0800.0000.126F  ;3221  NOP
U 126F. 0000.003C.0180.C800.0000.1270  ;3222  D[LONG]_CACHE.P   ; READ THE LOCATION
U 1270. 0000.003C.75F0.2C00.0000.1271  ;3223  Q_ID[MAINT]   ; READ THE MAINT REG
U 1271. 0001.203C.0180.09E0.0000.1272  ;3224  RC[0C]_Q
U 1272. 0010.0038.01C0.0970.0000.1273  ;3225  Q_RC[0E]   ; GET EXPECTED DATA
U 1273. 001D.2000.0180.0800.0010.1274  ;3226  ALU_Q-D,CLK.UBCC  ; CHECK IT
U 1274. 0001.013C.0180.09E8.0000.10B0  ;3227  Z?,RC[0D]_D
U 10B0. 0000.003D.0180.0800.0000.1109  ;3228 =0 ERROR1    ; READ DATA INCORRECT
          ;3229 VEND002:
U 10B1. 0000.003C.0180.0800.0000.10B2  ;3230  NOP
          ;3231
          ;3232
```

```
                ;3233 .PAGE "TEST 20E NO SBI INVALIDATE IN GR0 ON I/O WRITE CYCLE"
                ;3234 ;*****************************************************************
                ;3235 ;++
                ;3236 ;
                ;3237 ; NO SBI INVALIDATE IN GR0 ON I/O WRITE CYCLE
                ;3238 ;
                ;3239 ;TEST DESCRIPTION
                ;3240 ; THIS TEST CHECKS THAT THE CACHE IS NOT INVALIDATED ON AN
                ;3241 ; I/O WRITE CYCLE. THE CACHE IS VALIDATED WITH KNOWN DATA.
                ;3242 ; THE CORRESPONDING I/O ADDRESS (BIT 29 SET) IS THEN WRITTEN
                ;3243 ; WITH DIFFERENT DATA. THE ORIGINAL LOCATION IS THEN READ AND
                ;3244 ; THE MAINTENANCE REGISTER IS CHECKED TO SEE IF THE LOCATION
                ;3245 ; WAS A HIT IN GROUP 0.
                ;3246 ; THE ADDRESSES USED ARE 2004 AND 20002004.
                ;3247 ;
                ;3248 ;LOGIC DESCRIPTION
                ;3249 ; THE WRITE INVALID LOGIC ON SBHM AND ASSOCIATED LOGIC ON
                ;3250 ; SBH ARE CHECKED.
                ;3251 ;
                ;3252 ;ERROR DESCRIPTION
                ;3253 ; EXPECTED HIT/MISS BIT <09>
                ;3254 ; RECEIVED HIT/MISS BIT <09>
                ;3255 ;
                ;3256 ;--
                ;3257 ;*****************************************************************
                ;3258 =0
U 10B2, 0018,0039,0D80,09F8,0000,10EA  ;3259 IOINV0: NEWTST[.3]
U 10B3, 0000,003C,0180,0800,0000,10B4  ;3260  NOP
U 10B4, 0000,003D,0180,0800,0000,1046  ;3261 =0 CALL,J/MAINTREGSETUP  ;SETUP SOME REGISTERS
U 10B5, 0000,003C,6580,0800,0084,7275  ;3262  SC_K[.10]
U 1275, 0818,0038,A5F8,0800,0000,1276  ;3263  D_K[.60],Q_0   ;INIT THE SBI INVALIDATE BITS IN
U 1276, 0D00,003C,0180,0800,0000,1277  ;3264  D_DAL.SC   ;THE MAINTENANCE REGISTER
U 1277, 0001,003C,0180,0988,0000,1278  ;3265  RC[1]_D    ;SAVE
U 1278, 0000,003C,01C0,0A70,0000,1279  ;3266  Q_R[0E]    ;I/O ADDRESS TO BE USED
U 1279, 0818,0038,B180,0800,0000,10B6  ;3267  D_K[.3FF]
U 10B6, 0000,003D,0180,0800,0000,112B  ;3268 =0 CALL,J/DCF
U 10B7, 001D,0034,0180,0A98,0000,10B8  ;3269  R[3]_D.AND.Q   ;INIT ADDRESS TO BE VALIDATED
U 10B8, 0000,003D,0180,0800,0000,10FF  ;3270 =0 ERLOOP
U 10B9, 0000,003C,0180,0A18,0200,127A  ;3271 STRT: VA_R[3]    ;LOAD ADDRESS TO VALIDATE
U 127A, 0818,0038,0580,0800,0000,127B  ;3272  D_K[.1]    ;GET DATA PATTERN TO WRITE
U 127B, 0000,003C,0180,9800,0000,127C  ;3273  MCT/VALIDATE   ;VALIDATE THE LOCATION
U 127C, 0000,003C,01C0,0A30,0000,127D  ;3274  Q_R[6]    ;GET GR0 BITS
U 127D, 0810,0038,0180,0908,0000,127E  ;3275  D_RC[1]
U 127E, 081D,0030,0180,0800,0000,127F  ;3276  D_D.OR.Q   ;COMBINE WITH INVALIDATE BITS
U 127F, 0000,003C,7580,3C00,0000,1280  ;3277  ID[MAINT]_D   ;SET MAINT REGISTER
U 1280, 0F00,003C,0180,0A70,0200,1281  ;3278  VA_R[0E],D_0   ;LOAD I/O ADDRESS AND DATA
U 1281, 0000,003C,0180,A800,0000,1282  ;3279  MCT/WRITE.P   ;WRITE TO I/O
U 1282, 0018,0038,11C0,0800,0000,1283  ;3280  Q_K[.4]
U 1283, 0019,2000,05C0,0800,0010,1284  ;3281 CNTNP0: Q_Q-K[.1],CLK.UBCC  ;WAIT FOR WRITE TO COMPLETE
U 1284, 0000,013C,0180,0800,0000,10BA  ;3282  Z?
U 10BA, 0000,003C,0180,0800,0000,1283  ;3283 =0 J/CNTNP0
U 10BB, 0000,003C,0180,0A18,0200,1285  ;3284  VA_R[3]    ;LOAD ORIGINAL ADDRESS
U 1285, 0000,003C,0180,C800,0000,1286  ;3285  D[LONG]_CACHE.P   ;READ THE LOCATION
U 1286, 0000,003C,75F0,2C00,0000,1287  ;3286  Q_ID[MAINT]   ;READ THE MAINT REG FOR CACHE HIT
U 1287, 0800,003C,0180,0A40,0000,1288  ;3287  D_R[8]    ;GET GR0 HIT BIT
```

```
U 1288. 0001.003C.0180.09F0.0000.1289  ;3288  RC[0E]_D   ;SAVE
U 1289. 081D.0034.0180.0800.0010.128A  ;3289  D_D.AND.Q,CLK.UBCC
U 128A. 0001.013C.0180.09E8.0000.10BC  ;3290  Z?,RC[0D]_D
U 10BC. 0000.003C.0180.0800.0000.128B  ;3291 =0 J/NOMEM    ;HIT BIT SET,GO TO END OF TEST
U 10BD. 0000.003D.0180.0800.0000.110B  ;3292  ERROR2     ;HIT BIT NOT SET
U 128B. 0000.003C.0180.0800.0000.10BE  ;3293 NOMEM: NOP
          ;3294
```

```
          ;3295 .PAGE "TEST 20F NO SBI INVALIDATE IN GR1 ON I/O WRITE CYCLE"
          ;3296 ;******************************************************************
          ;3297 ;++
          ;3298 ;
          ;3299 ; NO SBI INVALIDATE IN GR1 ON I/O WRITE CYCLE
          ;3300 ;
          ;3301 ;TEST DESCRIPTION
          ;3302 ; THIS TEST CHECKS THAT THE CACHE IS NOT INVALIDATED ON AN
          ;3303 ; I/O WRITE CYCLE. THE CACHE IS VALIDATED WITH KNOWN DATA.
          ;3304 ; THE CORRESPONDING I/O ADDRESS (BIT 29 SET) IS THEN WRITTEN
          ;3305 ; WITH DIFFERENT DATA. THE ORIGINAL LOCATION IS THEN READ AND
          ;3306 ; THE MAINTENANCE REGISTER IS CHECKED TO SEE IF THE LOCATION
          ;3307 ; WAS A HIT IN GROUP 1.
          ;3308 ; THE ADDRESSES USED ARE 2004 AND 20002004.
          ;3309 ;
          ;3310 ;LOGIC DESCRIPTION
          ;3311 ; THE WRITE INVALID LOGIC ON SBHM AND ASSOCIATED LOGIC ON
          ;3312 ; SBH ARE CHECKED.
          ;3313 ;
          ;3314 ;ERROR DESCRIPTION
          ;3315 ; EXPECTED HIT/MISS BIT <10>
          ;3316 ; RECEIVED HIT/MISS BIT <10>
          ;3317 ;
          ;3318 ;--
          ;3319 ;******************************************************************
          ;3320 =0
U 10BE. 0018,0039,0D80,09F8,0000,10EA  ;3321 IOINV1: NEWTST[.3]
U 10BF. 0000,003C,0180,0800,0000,10C0  ;3322  NOP
U 10C0. 0000,003D,0180,0800,0000,1046  ;3323 =0 CALL,J/MAINTREGSETUP  ;SETUP SOME REGISTERS
U 10C1. 0818,0038,4580,0800,0000,128C  ;3324  D_K[.8000]   ;SETUP TO FORCE REPLACE GR1
U 128C. 0500,003C,8980,0800,0084,728D  ;3325  D_D.LEFT,SC_K[.D]  ;AND FORCE MISS GROUP 0
U 128D. 0801,001C,0180,0800,0000,128E  ;3326  D_D.ORNOT.MASK
U 128E. 0001,003C,7580,3EA0,0000,128F  ;3327  R[4]_D,ID[MAINT]_D  ;LOAD THE MAINTENANCE REGISTER
U 128F. 0000,003C,6580,0800,0084,7290  ;3328  SC_K[.10]   ;INIT THE SBI INVALIDATE BITS
U 1290. 0818,0038,A5F8,0800,0000,1291  ;3329  D_K[.60],Q_0   ;IN THE MAINTENANCE REGISTER
U 1291. 0D00,003C,0180,0800,0000,1292  ;3330  D_DAL.SC
U 1292. 0001,003C,0180,0988,0000,1293  ;3331  RC[1]_D    ;SAVE
U 1293. 0018,0038,0580,09F0,0000,1294  ;3332  RC[0E]_K[.1]   ;GENERATE EXPECTED DATA
U 1294. 0018,0038,0D80,09E0,0000,1295  ;3333  RC[0C]_K[.3]   ;SET THE LOOP COUNT
U 1295. 0000,003C,01C0,0A70,0000,1296  ;3334  Q_R[0E]    ;I/O ADDRESS TO BE USED
U 1296. 0818,0038,8180,0800,0000,10C2  ;3335  D_K[.3FF]
U 10C2. 0000,003D,0180,0800,0000,112B  ;3336 =0 CALL,J/DCF   ;INIT THE ADDRESS TO BE
U 10C3. 001D,0034,0180,0A98,0000,10C4  ;3337  R[3]_D.AND.Q   ;VALIDATED
U 10C4. 0000,003D,0180,0800,0000,10FF  ;3338 =0 ERLOOP
U 10C5. 0000,003C,0180,0A18,0200,1297  ;3339 STRT1: VA_R[3]    ;LOAD ADDRESS TO VALIDATE
U 1297. 0818,0038,0580,0800,0000,1298  ;3340  D_K[.1]    ;GET DATA PATTERN TO WRITE
U 1298. 0000,003C,0180,9800,0000,1299  ;3341  MCT/VALIDATE   ;VALIDATE THE LOCATION
U 1299. 0000,003C,01C0,0A20,0000,129A  ;3342  Q_R[4]    ;GET GR1 BITS
U 129A. 0810,0038,0180,0908,0000,129B  ;3343  D_RC[1]    ;COMBINE WITH INVALIDATE BITS
U 129B. 081D,0030,0180,0800,0000,129C  ;3344  D_D.OR.Q
U 129C. 0000,003C,7580,3C00,0000,129D  ;3345  ID[MAINT]_D   ;SET MAINTENANCE REGISTER
U 129D. 0F00,003C,0180,0A70,0200,129E  ;3346  VA_R[0E],D_0   ;LOAD I/O ADDRESS AND DATA
U 129E. 0000,003C,0180,A800,0000,129F  ;3347  MCT/WRITE.P   ;WRITE TO I/O
U 129F. 0018,0038,11C0,0800,0000,12A0  ;3348  Q_K[.4]    ;WAIT FOR WRITE TO COMPLETE
U 12A0. 0019,2000,05C0,0800,0010,12A1  ;3349 CNTNP1: Q_Q-K[.1],CLK.UBCC
```

```
U 12A1. 0000.013C.0180.0800.0000.10C6  ;3350  Z?
U 10C6. 0000.003C.0180.0800.0000.12A0  ;3351 =0 J/CNTNP1
U 10C7. 0000.003C.0180.0A18.0200.12A2  ;3352  VA_R[3]    ;LOAD ORIGINAL ADDRESS
U 12A2. 0000.003C.0180.C800.0000.12A3  ;3353  D[LONG]_CACHE.P    ;READ THE LOCATION
U 12A3. 0000.003C.75F0.2C00.0000.12A4  ;3354  Q_ID[MAINT]    ;READ THE MAINT REG FOR CACHE HIT
U 12A4. 0800.003C.0180.0A48.0000.12A5  ;3355  D_R[9]    ;GET GR1 HIT BIT
U 12A5. 0001.003C.0180.09F0.0000.12A6  ;3356  RC[0E]_D    ;SAVE
U 12A6. 081D.0034.0180.0800.0010.12A7  ;3357  D_D.AND.Q,CLK.UBCC
U 12A7. 0001.013C.0180.09E8.0000.10C8  ;3358  Z?,RC[0D]_D
U 10C8. 0000.003C.0180.0800.0000.12A8  ;3359 =0 J/NOMEM1    ;HIT BIT SET, GO TO END OF TEST
U 10C9. 0000.003D.0180.0800.0000.110B  ;3360  ERROR2    ;HIT BIT NOT SET
U 12A8. 0000.003C.0180.0800.0000.10CA  ;3361 NOMEM1: NOP
            ;3362
            ;3363
```

```
                ;3364 .PAGE "TEST 210 INTERRUPT SUMMARY READ TEST"
                ;3365 ;*****************************************************************
                ;3366 ;**
                ;3367 ;
                ;3368 ; INTERRUPT SUMMARY READ TEST
                ;3369 ;
                ;3370 ;TEST DESCRIPTION
                ;3371 ; THIS TEST CHECKS THAT THE CPU RECEIVES THE CORRECT RESPONSE
                ;3372 ; FROM DEVICES WHEN AN SBI INTERRUPT OCCURS AND THE CPU ISSUES
                ;3373 ; AN INTERRUPT SUMMARY READ COMMAND.
                ;3374 ; THE SBI IS SEARCHED FOR DEVICES. IF A DEVICE IS FOUND, THE
                ;3375 ; DEVICE CODE IS CHECKED TO DETERMINE WHETHER IT IS A UBA OR
                ;3376 ; AN MBA. WHEN THAT DETERMINATION IS MADE, EXECUTION GOES TO
                ;3377 ; THE APPROPRIATE ROUTINE TO CAUSE THAT DEVICE TO INTERRUPT.
                ;3378 ; ONCE THE INTERRUPT IS ACTIVE, THE CPU ISSUES AN INTERRUPT
                ;3379 ; SUMMARY READ COMMAND AND WAITS FOR THE DEVICE TO RESPOND.
                ;3380 ; THE RESPONSE IS CHECKED FOR THE CORRECT BIT PAIR FOR THE
                ;3381 ; CURRENT TR BEING TESTED.
                ;3382 ;
                ;3383 ; THE UBA ROUTINE SETS AND CLEARS THE UNIBUS POWER DOWN BIT
                ;3384 ; IN UBA CONTROL REGISTER 1 WITH THE INTERRUPT ENABLE BIT
                ;3385 ; SET. THIS WILL CAUSE THE UBA TO INTERRUPT.
                ;3386 ;
                ;3387 ; THE MBA ROUTINE SETS THE MAINTENANCE MODE AND INTERRUPT
                ;3388 ; ENABLE BITS IN MBA REGISTER 1 AND THEN SETS AND CLEARS THE
                ;3389 ; SIMULATE ATTENTION BIT IN REGISTER 4. THIS WILL CAUSE AN
                ;3390 ; INTERRUPT.
                ;3391 ;
                ;3392 ;LOGIC DESCRIPTION
                ;3393 ; THE ISR LOGIC ON SBLA AND ASSOCIATED LOGIC ON M8218
                ;3394 ; AND M8219 IS TESTED.
                ;3395 ;
                ;3396 ;ERROR DESCRIPTION
                ;3397 ; EXPECTED INTERRUPT SUMMARY RESPONSE BIT PAIR
                ;3398 ; RECEIVED INTERRUPT SUMMARY RESPONSE BIT PAIR
                ;3399 ; UBA/MBA FLAG        0=UBA, 1=MBA
                ;3400 ;
                ;3401 ;--
                ;3402 ;*****************************************************************
                ;3403 =0
U 10CA. 0000.003D.0180.0800.0000.10EA  ;3404 ISRTST: NEWTST
U 10CB. 0000.003C.0180.0800.0000.10CC  ;3405  NOP
U 10CC. 0000.003D.0180.0800.0000.1046  ;3406 =0 CALL,J/MAINTREGSETUP
U 10CD. 0018.0038.0D80.09F8.0000.12A9  ;3407  RC[0F]_K[.3]
U 12A9. 0F00.003C.0180.0800.0000.12AB  ;3408  D_0
U 12AB. 0000.003C.3580.3C00.0000.12AC  ;3409  ID[VECTOR]_D   ;CLEAR VECTOR REGISTER
U 12AC. 0000.003C.3D80.3C00.0000.10CE  ;3410  ID[PSL]_D    ;CLEAR PSL
U 10CE. 0000.003D.0180.0800.0000.11BD  ;3411 =0 CALL,J/LOOKFORDEVICE  ;TRY TO FIND A UBA
U 10CF. 0000.003C.0180.0A00.0210.12AD  ;3412 CHECK: VA_ALU,ALU_R[0],CLK.UBCC ;WAS THERE A DEVICE?
U 12AD. 0000.013C.0180.0800.0000.10D0  ;3413  Z?
U 10D0. 0000.003C.0180.0800.0000.12AE  ;3414 =0 J/CHECKCODEISR    ;DEVICE FOUND UBA?
U 10D1. 0000.003C.0180.0800.0000.12E7  ;3415  J/ENTST     ;NO DEVICE AVAILABLE
                ;3416 CHECKCODEISR:
U 12AE. 0818.003B.6580.0800.0000.12AF  ;3417  D_K[.10]
U 12AF. 0000.003C.01C0.0A10.0000.12B0  ;3418  Q_R[2]
```

```
U 12B0. 081D.0000.0180.0800.0000.12B1  ;3419  D_D-Q    ;GET TR OF DEVICE
U 12B1. 0001.003C.0180.0800.0082.12B2  ;3420  SC_D
U 12B2. 0803.001C.0180.0800.0000.12B3  ;3421  D_NOT.MASK   ;GENERATE EXPECTED BIT PAIR
U 12B3. 0000.003C.6580.0800.0084.92B4  ;3422  SC_SC+K[.10]   ;FOR THE INT.SUM. RESPONSE
U 12B4. 0801.001C.0180.0800.0000.12B5  ;3423  D_D.ORNOT.MASK
U 12B5. 0001.003C.0180.09F0.0000.12B6  ;3424  RC[0E]_D   ;SAVE
U 12B6. 0000.003C.0180.C800.0000.12B7  ;3425  D[LONG]_CACHE.P   ;READ THE CONFIGURATION REG
U 12B7. 0200.003C.0180.0800.0000.12B8  ;3426  D_D.RIGHT2
U 12B8. 0600.003C.0180.0800.0000.12B9  ;3427  D_D.RIGHT   ;CHECK BITS 3 AND 4
U 12B9. 0000.193C.0180.0800.0000.106C  ;3428  D1-0?    ;UBA=01
U 106C. 0000.003C.0180.0800.0000.12C6  ;3429 =1100 J/MBSETUP   ;SETUP FOR MBA INTERRUPT
U 106D. 0018.0038.1980.09E0.0000.106E  ;3430  RC[0C]_K[ZERO]   ;SET UBA FLAG
U 106E. 0800.003C.0180.0A00.0000.106F  ;3431  D_R[0]
U 106F. 0019.0014.1180.0800.0200.12BA  ;3432  VA_ALU,ALU_D+K[.4]  ;LOAD ADDR OF REG 1
U 12BA. 0818.0038.0980.0800.0000.12BB  ;3433  D_K[.2]
U 12BB. 0000.003C.0180.A800.0000.12BC  ;3434  CACHE.P_D[LONG]   ;SET POWER FAIL BIT
U 12BC. 0818.0038.1180.0800.0000.12BD  ;3435  D_K[.4]
U 12BD. 0000.003C.0180.A800.0000.10D2  ;3436  CACHE.P_D[LONG]   ;CLR POWERFAIL,SET IE
U 10D2. 0000.003D.0180.0800.0000.11E1  ;3437 =0 CALL,J/WAITFORUNIBUS  ;WAIT FOR UBA TO POWER UP
U 10D3. 0000.003C.0180.0800.0000.10D4  ;3438  NOP
U 10D4. 0000.003D.0180.0800.0000.10FF  ;3439 =0 ERLOOP
U 10D5. 0000.003C.0180.0800.4000.12BE  ;3440  INTRPT.STROBE   ;LATCH THE INTERRUPT
U 12BE. 0000.003C.0180.0800.0000.12BF  ;3441  NOP
U 12BF. 0000.003C.0180.D800.0000.12C0  ;3442  MCT/READ.INT.SUM  ;DO INTERRUPT SUM READ
U 12C0. 0000.003C.0180.0800.0000.12C1  ;3443  NOP    ;WAIT FOR RESPONSE
U 12C1. 0000.003C.0180.0800.0000.12C2  ;3444  NOP
U 12C2. 0001.003C.0180.09E8.0000.12C3  ;3445  RC[0D]_D   ;SAVE INTERRUPT SUM RESPONSE
U 12C3. 0010.0038.01C0.0970.0000.12C4  ;3446  Q_RC[0E]   ;GET EXPECTED DATA
U 12C4. 001D.0000.0180.0800.0010.12C5  ;3447  ALU_D-Q,CLK.UBCC
U 12C5. 0000.013C.0180.0800.0000.10D6  ;3448  Z?
U 10D6. 0000.003D.0180.0800.0000.110B  ;3449 =0 ERROR2    ;RESPONSE DATA INCORRECT
U 10D7. 0000.003C.0180.0800.0000.10E0  ;3450  J/NXTR    ;CHECK FOR ANOTHER DEVICE
        ;3451
        ;3452 MBSETUP:
U 12C6. 0018.0038.0580.09E0.0000.12C7  ;3453  RC[0C]_K[.1]   ;SET MBA FLAG
U 12C7. 0800.003C.0180.0A00.0000.12C8  ;3454  D_R[0]
U 12C8. 0019.0014.1180.0800.0200.12C9  ;3455  VA_ALU,ALU_D+K[.4]  ;LOAD ADDR OF REG 1
U 12C9. 0818.0038.0580.0800.0000.10D8  ;3456  D_K[.1]
U 10D8. 0000.003D.0180.A800.0000.11E1  ;3457 =0 CACHE.P_D[LONG],CALL[WAITFORUNIBUS] ;INIT THE MBA
U 10D9. 0818.0038.F980.0803.0000.12CA  ;3458  VA_VA+4,D_K[.7E]  ;LOAD ADDR OF REG 2
U 12CA. 0819.0034.6180.0800.0000.12CB  ;3459  D_D.AND.K[.F]
U 12CB. 0000.003C.65F8.0800.0084.72CC  ;3460  SC_K[.10],Q_0
U 12CC. 0D00.003C.0180.0800.0000.12CD  ;3461  D_DAL.SC   ;SETUP TO CLR BITS 17,18,19
U 12CD. 0000.003C.0180.A800.0000.12CE  ;3462  CACHE.P_D[LONG]
U 12CE. 0800.003C.0180.0A00.0000.12CF  ;3463  D_R[0]
U 12CF. 0019.0014.1180.0800.0200.12D0  ;3464  VA_ALU,ALU_D+K[.4]  ;LOAD ADDR OF REG 1
U 12D0. 0818.0038.8580.0800.0000.12D1  ;3465  D_K[.C]
U 12D1. 0000.003C.0180.A800.0000.12D2  ;3466  CACHE.P_D[LONG]   ;SET MMODE AND IE BITS
U 12D2. 0800.003C.2180.0A00.0000.12D3  ;3467  D_R[0],K[.14]
U 12D3. 0019.0014.2180.0800.0200.12D4  ;3468  VA_ALU,ALU_D+K[.14]  ;LOAD ADDR OF REG 5
U 12D4. 0818.0038.0580.0800.0000.12D5  ;3469  D_K[.1]
U 12D5. 0B00.003C.0180.0800.0000.12D6  ;3470  D_D.SWAP
U 12D6. 0000.003C.0180.A800.0000.12D7  ;3471  CACHE.P_D[LONG]   ;SET SIMULATE ATTENTION
U 12D7. 0818.0038.6580.0800.0000.12D8  ;3472  D_K[.10]
U 12D8. 0819.0000.0580.0800.0010.12D9  ;3473 WAITMBA:D_D-K[.1],CLK.UBCC  ;WAIT FOR THE INTERRUPT
```

```
U 12D9. 0000.013C.0180.0800.0000.10DA  ;3474  Z?
U 10DA. 0000.003C.0180.0800.0000.12D8  ;3475 =0 J/WAITMBA
U 10DB. 0000.003C.0180.0800.0000.10DC  ;3476  NOP
U 10DC. 0000.003D.0180.0800.0000.10FF  ;3477 =0 ERLOOP
U 10DD. 0000.003C.0180.0800.4000.12DA  ;3478  INTRPT.STROBE   ;LATCH THE INTERRUPT
U 12DA. 0000.003C.0180.D800.0000.12DB  ;3479  MCT/READ.INT.SUM  ;DO INTERRUPT SUM READ
U 12DB. 0000.003C.0180.0800.0000.12DC  ;3480  NOP    ;WAIT FOR RESPONSE
U 12DC. 0000.003C.0180.0800.0000.12DD  ;3481  NOP
U 12DD. 0001.003C.0180.09E8.0000.12DE  ;3482  RC[0D]_D   ;SAVE INTERRUPT SUM RESPONSE
U 12DE. 0010.0038.01C0.0970.0000.12DF  ;3483  Q_RC[0E]
U 12DF. 001D.0000.0180.0800.0010.12E0  ;3484  ALU_D-Q,CLK.UBCC
U 12E0. 0000.013C.0180.0800.0000.10DE  ;3485  Z?
U 10DE. 0000.003D.0180.0800.0000.110B  ;3486 =0 ERROR2    ;RESPONSE DATA INCORRECT
U 10DF. 0000.003C.0180.0800.0000.10E0  ;3487  NOP
          ;3488 =0
U 10E0. 0000.003D.0180.0800.0000.1144  ;3489 NXTR: CALL,J/UNJAM   ;CLEAR THE INTERRUPTS
U 10E1. 0F00.003C.0180.0800.0000.12E1  ;3490  D_0
U 12E1. 0000.003C.3580.3C00.0000.12E2  ;3491  ID[VECTOR]_D
U 12E2. 0000.003C.7580.3C00.0000.12E3  ;3492  ID[MAINT]_D
U 12E3. 0000.003C.3D80.3C00.0000.12E4  ;3493  ID[PSL]_D
U 12E4. 0000.003C.0180.0800.0084.72E5  ;3494  SC_K[.8]
U 12E5. 0818.0038.75F8.0800.0000.12E6  ;3495  D_K[.20],Q_0
U 12E6. 0D00.003C.0180.0800.0000.10E2  ;3496  D_DAL.SC
U 10E2. 0000.003D.89E0.0800.0084.707F  ;3497 =0 Q_D,SC_K[.D],CALL,J/FOUNDMEMORY  ;CHECK FOR NEXT TR
U 10E3. 0000.003C.0180.0800.0000.10CF  ;3498  J/CHECK
U 12E7. 0000.003C.0180.0800.0000.10E4  ;3499 ENTST: NOP
          ;3500
```

```
            ;3501  .PAGE "TEST 211 RESERVED"
            ;3502 =0
U 10E4, 0000,003D,0180,0800,0000,10EA  ;3503 T211: NEWTST
U 10E5, 0000,003C,0180,0800,0000,10E6  ;3504  NOP
            ;3505
```

```
            ;3506  .PAGE "TEST 212 RESERVED"
            ;3507 =0
U 10E6, 0000,003D,0180,0800,0000,10EA  ;3508 T212: NEWTST
U 10E7, 0000,003C,0180,0800,0000,12E8  ;3509  NOP
            ;3510
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 2121
ESKAR56.MCR                    MICRO2  1P(00)    26-JUL-85  17:09:57          Page   9

            ;3511  .PAGE "DUMMY NEWTST"
U 12E8. 0000,003D,0180,0800,0000,10EA  ;3512 FAKE: NEWTST
            ;3513
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2122
ESKAR56.MCR                    MICRO2  1P(00)    26-JUL-85  17:09:57     Page   9
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1911: 1880 1882= 1901= 2428= 2429= 2430= 1881
U 1008   2517= 2518= 2519= 2520= 2521= 2522= 2523= 2524=
U 1010   2435= 2436= 2437= 1904 2441= 2442= 2443= 1905
U 1018   2904= 2905= 2906= 2907= 2908= 2909= 2910= 2911=
U 1020   2447= 2450= 2452= 2453= 1902= 1903= 1906 2944=
U 1028   2799= 2800= 2801= 2803= 1919= 1920= 1907 2945=
U 1030   2848= 2849= 2850= 2852= 1973= 1975= 1986= 1987=
U 1038   2008= 2009= 2032= 2033= 3004= 3005= 2137= 2138=
U 1040   2297= 2299= 2302= 2304= 2307= 2309= 2336= 2337=
U 1048   2338= 2339= 2348= 2349= 2351= 2352= 2014: 1908
U 1050   2353= 2354= 2494= 2495= 2496= 2497= 2505= 2506=
U 1058   2507= 2509= 2038: 1909 2512= 2513= 2042: 1910
U 1060   2554= 2555= 2556= 2557= 2564= 2565= 2571= 2573=
U 1068   2576= 2577= 2597= 2598= 3429= 3430= 3431= 3432=
U 1070   2599= 2600= 2602= 2603= 2695= 2696= 2699= 2700=
U 1078   2711= 2712= 2715= 2716= 2753= 2754= 2881= 2883=
U 1080   2892= 2893= 2926= 2927= 2988= 2989= 2990= 2991=
U 1088   2992= 2994= 2996= 2997= 3019= 3020= 3024= 3025=
U 1090   3041= 3042= 3067= 3068= 3075= 3076= 3079= 3080=
U 1098   3081= 3082= 3084= 3085= 3086= 3087= 3117= 3118=
U 10A0   3119= 3120= 3131= 3132= 3143= 3144= 3153= 3155=
U 10A8   3188= 3189= 3190= 3191= 3206= 3207= 3218= 3219=
U 10B0   3228= 3230= 3259= 3260= 3261= 3262= 3268= 3269=
U 10B8   3270= 3271= 3283= 3284= 3291= 3292= 3321= 3322=
U 10C0   3323= 3324= 3336= 3337= 3338= 3339= 3351= 3352=
U 10C8   3359= 3360= 3404= 3405= 3406= 3407= 3411= 3412=
U 10D0   3414= 3415= 3437= 3438= 3439= 3440= 3449= 3450=
U 10D8   3457= 3458= 3475= 3476= 3477= 3478= 3486= 3487=
U 10E0   3489= 3490= 3497= 3498= 3503= 3504= 3508= 3509=
U 10E8   1917 1918 1962 1963 1964 1965 1966 1967
U 10F0   1968 1969 1970 1971 1972 1976 1977 1978
U 10F8   1979 1980 1981 1982 1983 1984 1985 1989
U 1100   1866: 1867: 1868: 1869: 1870: 1871: 1872: 1873:
U 1108   1874: 2012 2013 2036 1875: 1876: 1877: 1878:
U 1110   2037 2090 2091 2092 2111 2112 2113 2114
U 1118   2139 2140 2141 2142 2168 2169 2170 2171
U 1120   2172 2173 2174 2175 2176 2177 2178 2179
U 1128   2180 2181 2182 2183 2185 2186 2187 2188
U 1130   2190 2200 2201 2224 2233 2234 2235 2241
U 1138   2269 2270 2271 2273 2274 2275 2276 2278
U 1140   2279 2280 2281 2282 2292 2293 2294 2295
U 1148   2296 2300 2301 2305 2306 2310 2311 2340
U 1150   2341 2345 2347 2350 2355 2236: 2356 2357
U 1158   2358 2359 2360 2361 2362 2363 2364 2365
U 1160   2366 2367 2368 2369 2370 2371 2372 2373
U 1168   2374 2375 2376 2377 2378 2379 2380 2381
U 1170   2426 2427 2454 2498 2499 2500 2501 2510
U 1178   2511 2515 2516 2529 2530 2532 2533 2534
U 1180   2536 2541 2542 2543 2545 2550 2552 2553
U 1188   2561 2562 2566 2567 2568 2569 2570 2578
U 1190   2579 2580 2582 2583 2584 2585 2586 2587
U 1198   2593 2595 2596 2601 2626 2627 2628 2629
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2659 2660 2662 2663 2664 2691 2693 2694
U 11A8  2698 2702 2703 2704 2705 2707 2708 2709
U 11B0  2710 2713 2714 2742 2743 2744 2745 2746
U 11B8  2747 2748 2749 2751 2752 2861 2862 2863
U 11C0  2864 2865 2866 2867 2868 2869 2870 2871
U 11C8  2872 2873 2874 2876 2877 2878 2879 2880
U 11D0  2884 2885 2886 2887 2888 2889 2890 2891
U 11D8  2895 2896 2897 2898 2900 2901 2902 2903
U 11E0  2913 2922 2923 2924 2925 2936 2938 2939
U 11E8  2940 2941 2942 2943 2946 2948 2949 2950
U 11F0  2952 2957 2958 2959 2995 3000 3001 3002
U 11F8  3003 3006 3007 3008 3009 3010 3011 3012
U 1200  3013 3016 3017 3018 3026 3027 3028 3029
U 1208  3030 3031 3032 3033 3034 3035 3036 3037
U 1210  3038 3039 3040 3043 3044 3045 3046 3047
U 1218  3048 3049 3050 3051 3052 3053 3054 3055
U 1220  2243: 3056 3057 3058 2244: 3059 3060 3061
U 1228  2245: 3062 3063 3064 2246: 3065 3066 3069
U 1230  2247: 3070 3071 3072 2248: 3073 3074 3121
U 1238  2249: 3122 3123 3124 2250: 3125 3126 3127
U 1240  3128 3129 3130 3133 3134 3135 3136 3137
U 1248  3138 3139 3140 3141 3142 3145 3146 3147
U 1250  3148 3149 3150 3151 3152 3192 3193 3194
U 1258  3195 3196 3197 3198 3199 3200 3201 3202
U 1260  3203 3204 3205 3208 3209 3210 3211 3212
U 1268  3213 3214 3215 3216 3217 3220 3221 3222
U 1270  3223 3224 3225 3226 3227 3263 3264 3265
U 1278  3266 3267 3272 3273 3274 3275 3276 3277
U 1280  3278 3279 3280 3281 3282 3285 3286 3287
U 1288  3288 3289 3290 3293 3325 3326 3327 3328
U 1290  3329 3330 3331 3332 3333 3334 3335 3340
U 1298  3341 3342 3343 3344 3345 3346 3347 3348
U 12A0  3349 3350 3353 3354 3355 3356 3357 3358
U 12A8  3361 3408 2237: 3409 3410 3413 3417 3418
U 12B0  3419 3420 3421 3422 3423 3424 3425 3426
U 12B8  3427 3428 3433 3434 3435 3436 3441 3442
U 12C0  3443 3444 3445 3446 3447 3448 3453 3454
U 12C8  3455 3456 3459 3460 3461 3462 3463 3464
U 12D0  3465 3466 3467 3468 3469 3470 3471 3472
U 12D8  3473 3474 3479 3480 3481 3482 3483 3484
U 12E0  3485 3491 3492 3493 3494 3495 3496 3499
U 12E8  3512
U 12F0 - 13EF Unused
U 13F0  2048: 2049: 2050: 2051: 2052: 2053: 2054: 2055:
U 13F8  2056: 2057: 2058: 2059: 2060: 2061: 2062: 2063:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR56	761
Used	761
Remaining	

Total microwords used in memory U: 761
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR56.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2126
ESKAR57.MCR MICRO2 1P(00) 4-NOV-85 13

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1939 Micro Monitor Interface Routines
; 1940   Newtest Routine
; 1999   ERROR1 WITH PARAMETER
; 2025   ERROR 2 WITH PARAMETER
; 2054   Disable Cache Routine
; 2147   RESET SUBTEST NUMBER
; 2170   Set Trap Mask
; 2493   Enable SBI
; 2514   Initialize the SBI
; 2541   Wait Loop Routine
; 2573   Find MS780-E Memory
; 2716   Generate IO Address
; 2743   Set Starting Address
; 2779   ENABLE NEXT MS780-E
; 2831   CONFIGURE MS780-E/H
; 2905   MS780-E/H CNFG REG CLEANUP
; 2977   Select Controller
; 3016   SELECT LOWER CONTROLLER
; 3065   SELECT UPPER CONTROLLER
; 3114   SET UP MS780-E RDS
; 3178   SET UP MS780-E CRD
; 3243 TEST 213 NEXUS CRD/RDS TEST
; 3892 TEST 214 RESERVED
; 3897 TEST 215 RESERVED
; 3902 TEST 216 RESERVED
; 3907 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR57
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ; Test 213  Edit # 02-02
;1836 ; Added subroutine "INIT_DW780" and inserted calls to this subroutine
;1837 ; in subtests 3 and 6 to initialize the DW780 before using it to do
;1838 ; a read from memory. Also, in both subtests, changed macro ALU_D.AND.MASK
;1839 ; to ALU_D.ANDNOT.MASK so that the expected satus register bit(bit 9
;1840 ; for subt.3 and bit 8 for subt.6) would get masked in instead of
;1841 ; masked out.
;1842 ;
;1843 ;
;1844 ;
;1845 ;*****************************************************************************
;1846
```

```
                ;1847  .PAGE
                ;1848
                ;1849 ;*
                ;1850 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1851 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
                ;1852 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1853 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1854 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1855 ;
                ;1856 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1857 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1858 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1859 ; R[0F] AND DO A RETURN0.
                ;1860 ;
                ;1861 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1862 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1863 ; TO THE CONSOLE.
                ;1864 ;
                ;1865 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
                ;1866 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
                ;1867 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
                ;1868 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
                ;1869 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
                ;1870 ;-
                ;1871
U 1100, 0000,003C,19C0,0A78,0084,7001   ;1872 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101, 0000,003C,05C0,0A78,0084,7001   ;1873 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000,003C,09C0,0A78,0084,7001   ;1874 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103, 0000,003C,0DC0,0A78,0084,7001   ;1875 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104, 0000,003C,11C0,0A78,0084,7001   ;1876 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105, 0000,003C,D5C0,0A78,0084,7001   ;1877 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106, 0000,003C,D9C0,0A78,0084,7001   ;1878 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000,003C,5DC0,0A78,0084,7001   ;1879 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108, 0000,003C,01C0,0A78,0084,7001   ;1880 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C, 0000,003C,85C0,0A78,0084,7001   ;1881 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D, 0000,003C,89C0,0A78,0084,7001   ;1882 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E, 0000,003C,65C0,0A78,0084,7001   ;1883 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F, 0000,003C,61C0,0A78,0084,705B   ;1884 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
                ;1885
U 1001, 0001,2024,0180,0800,0010,1002   ;1886 TRPH0: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1002, 0000,013C,0180,0800,0000,1004   ;1887  Z?    ; BRANCH IF NO
U 1004, 0001,2036,0180,0AF8,0000,0000   ;1888 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1889
                ;1890
                ;1891 ;*
                ;1892 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1893 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1894 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1895 ;
                ;1896 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1897 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1898 ;       FAULT STATUS REGISTER
                ;1899 ;       SBI ERROR REGISTER
                ;1900 ;       TIMEOUT ADDRESS REGISTER
                ;1901 ;       MAINTENANCE REGISTER
```

```
                ;1902 ;       CACHE PARITY REGISTER
                ;1903 ;       TB ERROR REGISTER 1
                ;1904 ;       TB ERROR REGISTER 0
                ;1905 ;-
                ;1906
U 1005, 0018.0038.1D80.09E8.0000.1014  ;1907 TRPH1: RC[0D]_SC
U 1014, 0000.003D.D980.0800.0084.718D  ;1908 =0 SETRCF[.9]
U 1015, 0000.003C.49F0.2C00.0000.1006  ;1909  Q_ID[TBER0]
U 1006, 0001.203C.4DF0.2DB0.0000.1013  ;1910  RC[6]_Q,Q_ID[TBER1]
U 1013, 0001.203C.65F0.2DB8.0000.1016  ;1911  RC[7]_Q,Q_ID[SBI.ERR]
U 1016, 0001.203C.6DF0.2DD8.0000.101B  ;1912  RC[0B]_Q,Q_ID[FAULT]
U 101B, 0001.203C.75F0.2DE0.0000.101E  ;1913  RC[0C]_Q,Q_ID[MAINT]
U 101E, 0001.203C.79F0.2DC8.0000.1033  ;1914  RC[9]_Q,Q_ID[PARITY]
U 1033, 0001.203C.69F0.2DC0.0000.104F  ;1915  RC[8]_Q,Q_ID[TIME.ADDR]
U 104F, 0001.203C.81F0.2DD0.0000.1000  ;1916  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.116A  ;1917 1000: RC[0E]_Q,ERROR1
                ;1918
                ;1919 ;*
                ;1920 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
                ;1921 ;-
                ;1922
U 105B, 0001.2024.0180.0800.0010.105F  ;1923 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 105F, 0000.013C.0180.0800.0000.101C  ;1924  Z?     ; BRANCH IF NO
U 101C, 0001.2036.0180.0AF8.0000.0001  ;1925 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1 ; RETURN
U 101D, 0000.003C.0180.0800.0000.1005  ;1926  J/TRPH1     ; REPORT UNEXPECTED TRAP
                ;1927
                ;1928
                ;1929 ;*
                ;1930 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1931 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1932 ;  NEWTST
                ;1933 ;  ERLOOP
                ;1934 ;  ERROR1
                ;1935 ;  ERROR2
                ;1936 ;
                ;1937
```

```
                ;1938  .PAGE
                ;1939  .TOC " Micro Monitor Interface Routines"
                ;1940  .TOC "    Newtest Routine"
                ;1941 ;**
                ;1942 ; FUNCTIONAL DESCRIPTION:
                ;1943 ;
                ;1944 ; This routine Initializes the following registers:
                ;1945 ;  PSL    to 1F
                ;1946 ;  CES    to 0
                ;1947 ;  ACC.CS    to disable accellerator
                ;1948 ;  SIR    to 0
                ;1949 ;  CLK.CS    to stop interval timer
                ;1950 ;  TBER0    to disable the TB
                ;1951 ;  SBI.MAINT to     0
                ;1952 ;  SBI.ERR    to 0
                ;1953 ;  FAULT    to 0
                ;1954 ;  COMP    to 0
                ;1955 ;  RC[0E]    to  0
                ;1956 ;  R[0F]    to 0
                ;1957 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1958 ; A SCOPE call is then made to the Monitor.
                ;1959 ;
                ;1960 ; CALLING CONSTRAINT:
                ;1961 ;
                ;1962 ; =0
                ;1963 ;
                ;1964 ; SIDE EFFECTS:
                ;1965 ;
                ;1966 ; D Reg is destroyed
                ;1967 ; SC is destroyed
                ;1968 ;--
U 1109, 0000,003C,65F8,0800,0084,7155  ;1969 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1155, 0818,0038,8D80,0800,0000,1156  ;1970  d_k[.1f]  ; D=1F
U 1156, 0D00,003C,0180,0800,0000,1157  ;1971  d_dal.sc  ; D=001F0000
U 1157, 0000,003C,3D80,3C00,0000,1158  ;1972  id[psl]_d  ; psl = 001F0000
U 1158, 0818,0038,0580,0800,0000,1159  ;1973  d_k[.1]  ; Clear the CES register
U 1159, 0D00,003C,0180,0800,0000,115A  ;1974  d_dal.sc  ; D = 00010000
U 115A, 0F00,003C,3180,3C00,0000,115B  ;1975  id[ces]_d,d_0  ; ces = 00010000
U 115B, 0000,003C,5D80,3C00,0000,115C  ;1976  id[acc.cs]_d  ; acc.cs = 0
U 115C, 0000,003C,3980,3C00,0000,115D  ;1977  id[sir]_d  ; sir = 0
U 115D, 0000,003C,2980,3C00,0000,115E  ;1978  id[clk.cs]_d  ; clk.cs = 0
U 115E, 0000,003C,4980,3C00,0000,1044  ;1979  id[tber0]_d  ; tber0 = 0
U 1044, 0000,003D,0180,0800,0000,11D6  ;1980 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1981  intrpt.strobe,  ; Strobe interrupts
U 1045, 0818,0038,C180,0800,4000,115F  ;1982      d_k[.ffff]  ; Setup to clear SBI error register and
U 115F, 0801,0028,6580,3C00,0000,1160  ;1983  id[sbi.err]_d,d_not.d ; and fault register
U 1160, 0F00,003C,6D80,3C00,0000,1161  ;1984  id[fault]_d,d_0  ; ...
U 1161, 0000,003C,6D80,3C00,0000,1162  ;1985  id[fault]_d  ; fault = 0
U 1162, 0000,003C,7580,3C00,0000,1163  ;1986  id[maint]_d  ; MAINT = 0
U 1163, 0000,003C,6580,3C00,0000,1164  ;1987  id[sbi.err]_d  ; sbi error reg = 0
U 1164, 0000,003C,7180,3C00,0000,1165  ;1988  id[comp]_d  ; comp reg = 0
U 1165, 0000,003C,E580,3C00,0000,1166  ;1989  id[T9]_d  ; Clear code for subroutine START.TEST
U 1166, 0001,003C,0180,09F0,0000,1167  ;1990  rc[0e]_d  ;
U 1167, 0001,003C,0180,0AF8,0000,1168  ;1991  r[0f]_d  ; Clear expected trap mask
U 1168, 0001,003C,0180,09F8,0000,1046  ;1992  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1046, 0000,003D,0180,0800,0000,116E  ;1993 =0 call[disable.cache] ; Disable the cache
U 1047, 0F03,003C,0180,09E8,0000,105E  ;1994  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
          ;1995
U 1169, 0818,0038,0980,0800,0000,105E  ;1996 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
          ;1997
```

```
            ;1998  .PAGE
            ;1999  .TOC "   ERROR1 WITH PARAMETER"
            ;2000 ;++
            ;2001 ; FUNCTIONAL DESCRIPTION:
            ;2002 ;
            ;2003 ; This subroutine takes  as parameter the number of arguments
            ;2004 ; to be printed to the console in the case of an error logout.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; =0
            ;2009 ;INPUTS:
            ;2010 ;
            ;2011 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2012 ;--
            ;2013 =0
            ;2014 ERR1.ARG:
U 1048, 0000,003D,0180,0800,0000,118D  ;2015  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1049, 0000,003C,0180,0800,0000,116A  ;2016  J/ERROR1  ; Go to ERROR1
            ;2017 ;
            ;2018 ;
            ;2019
U 116A, 0810,0038,0180,0978,0000,116B  ;2020 ERROR1: D_RC[0F]
U 116B, 0819,0034,4D80,0800,0000,104E  ;2021  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2022 104E: D_D.OR.K[.4],J/CALLCON
            ;2023
```

```
            ;2024  .PAGE
            ;2025  .TOC "   ERROR 2 WITH PARAMETER"
            ;2026 ;++
            ;2027 ; FUNCTIONAL DESCRIPTION:
            ;2028 ;
            ;2029 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2030 ; except that in this case after setting the number of arguments
            ;2031 ; too the console, control is transferred to routine ERROR2.
            ;2032 ;
            ;2033 ; INPUTS:
            ;2034 ;
            ;2035 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2036 ;
            ;2037 ;--
            ;2038 =0
            ;2039 ERR2.ARG:
U 104A, 0000,003D,0180,0800,0000,118D  ;2040  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 104B, 0000,003C,0180,0800,0000,116C  ;2041  J/ERROR2  ; Go to ERROR2
            ;2042 ;
            ;2043 ;
            ;2044
U 116C, 0810,0038,0180,0978,0000,116D  ;2045 ERROR2: D_RC[0F]
U 116D, 0819,0034,4D80,0800,0000,105A  ;2046  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2047 105A: D_D.OR.K[.6],J/CALLCON
            ;2048
            ;2049 105E:
            ;2050 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2051  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2052
```

```
            ;2053  .PAGE
            ;2054  .TOC "   Disable Cache Routine"
            ;2055 ;++
            ;2056 ; FUNCTIONAL DESCRIPTION:
            ;2057 ;
            ;2058 ; This routine disables cache hits by setting bits <16:15>
            ;2059 ; in the maintenance register.
            ;2060 ;
            ;2061 ; CALLING SEQUENCE:
            ;2062 ;
            ;2063 ; =0
            ;2064 ;
            ;2065 ; SIDE EFFECTS:
            ;2066 ;
            ;2067 ; D & Q Registers destroyed
            ;2068 ;--
            ;2069 DISABLE.CACHE:
U 116E. 0818.0038.4580.0800.0000.116F  ;2070  d_k[.8000]  ; ... and seed constant
U 116F. 0500.003C.0180.0800.0000.1170  ;2071  d_d.left  ; Generate final constant
U 1170. 0819.0030.4580.0800.0000.1171  ;2072  d_d.or.k[.8000]  ; ... = 00018000
U 1171. 0000.003E.7580.3C00.0000.0001  ;2073  id[maint]_d,return1 ; Disable cache and return
            ;2074
            ;2075 ;+
            ;2076 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2077 ;-
            ;2078
U 13F0. 0000.003C.0180.0800.0000.13F1  ;2079 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;2080 13F1: NOP
U 13F2. 0000.003C.0180.0800.0000.13F3  ;2081 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;2082 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;2083 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;2084 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;2085 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;2086 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;2087 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;2088 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;2089 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;2090 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;2091 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;2092 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;2093 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1172  ;2094 13FF: NOP
            ;2095
            ;2096
            ;2097 ;+
            ;2098 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2099 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2100 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2101 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2102 ;
            ;2103 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2104 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2105 ;
            ;2106 ;   =0 D_0,CALL,J/DC5
            ;2107 ;    NOP
```

```
            ;2108 ;  =0 CALL,J/DC5
            ;2109 ;-
            ;2110
U 1172. 0018.0038.1980.0800.0000.1182  ;2111 DC0: ALU_0(K),J/S00
U 1173. 0018.0038.1980.0800.0000.1183  ;2112 DC1: ALU_0(K),J/S01
U 1174. 0018.0038.1980.0800.0000.1184  ;2113 DC2: ALU_0(K),J/S10
U 1175. 0018.0038.1980.0800.0000.1185  ;2114 DC3: ALU_0(K),J/S11
U 1176. 0018.0038.0980.0800.0000.1182  ;2115 DC4: ALU_K[.2],J/S00
U 1177. 0018.0038.0980.0800.0000.1183  ;2116 DC5: ALU_K[.2],J/S01
U 1178. 0018.0038.0980.0800.0000.1184  ;2117 DC6: ALU_K[.2],J/S10
U 1179. 0018.0038.0980.0800.0000.1185  ;2118 DC7: ALU_K[.2],J/S11
U 117A. 0018.0038.0580.0800.0000.1182  ;2119 DC8: ALU_K[.1],J/S00
U 117B. 0018.0038.0580.0800.0000.1183  ;2120 DC9: ALU_K[.1],J/S01
U 117C. 0018.0038.0580.0800.0000.1184  ;2121 DCA: ALU_K[.1],J/S10
U 117D. 0018.0038.0580.0800.0000.1185  ;2122 DCB: ALU_K[.1],J/S11
U 117E. 0018.0038.C180.0800.0000.1182  ;2123 DCC: ALU_K[.FFFF],J/S00
U 117F. 0018.0038.C180.0800.0000.1183  ;2124 DCD: ALU_K[.FFFF],J/S01
U 1180. 0018.0038.C180.0800.0000.1184  ;2125 DCE: ALU_K[.FFFF],J/S10
U 1181. 0018.0038.C180.0800.0000.1185  ;2126 DCF: ALU_K[.FFFF],J/S11
            ;2127
U 1182. 0118.0038.1B80.0800.0000.1186  ;2128 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 1183. 0118.0038.0B80.0800.0000.1186  ;2129 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
U 1184. 0118.0038.0780.0800.0000.1186  ;2130 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 1185. 0118.0038.C380.0800.0000.1186  ;2131 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
            ;2132
U 1186. 0100.003E.0380.0800.0000.0001  ;2133 SX: D_D.LEFT2,SI/7,RETURN1   ; EXIT
            ;2134
            ;2135
            ;2136 ;*
            ;2137 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;2138 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;2139 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;2140 ; TYPING IT.
            ;2141 ;-
            ;2142
U 1187. 0810.0038.4D80.0978.0000.1188  ;2143 SUBTST: D_RC[0F],KMX/.FF00
U 1188. 0019.4016.4D80.09F8.0000.0001  ;2144  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;2145
```

```
            ;2146  .PAGE
            ;2147  .TOC "   RESET SUBTEST NUMBER"
            ;2148 ;++
            ;2149 ; FUNCTIONAL DESCRIPTION:
            ;2150 ;
            ;2151 ; Since some  of the tests will have to be restarted when two
            ;2152 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2153 ; the subtest counter to zero ( only for thoes tests that have
            ;2154 ; more than one subtest). This subroutine may also be necessary
            ;2155 ; when a test is being loop on.
            ;2156 ;
            ;2157 ; CALLING CONSTRAINT:
            ;2158 ;
            ;2159 ; =0
            ;2160 ; SIDE EFFECTS:
            ;2161 ;
            ;2162 ; D is destroyed
            ;2163 ;
            ;2164 ;--
            ;2165 RESET.SUBTST:
            ;2166  RC[OF]_0,   ; Reset subtest number
U 1189, 0003,003E,0180,09F8,0000,0001  ;2167   RETURN1   ; ...and return
            ;2168
```

```
            ;2169  .PAGE
            ;2170  .TOC "   Set Trap Mask"
            ;2171 ;++
            ;2172 ; FUNCTIONAL DESCRIPTION:
            ;2173 ;
            ;2174 ; This routine is used to set a bit in the Micro Trap Mask
            ;2175 ; R[0F].
            ;2176 ;
            ;2177 ; CALLING CONSTRAINT:
            ;2178 ;
            ;2179 ; =0
            ;2180 ;
            ;2181 ; INPUTS:
            ;2182 ;
            ;2183 ; SC - Contains bit number to set.
            ;2184 ;
            ;2185 ; OUTPUTS:
            ;2186 ;
            ;2187 ; rc[0E] - Contains the current trap mask
            ;2188 ;
            ;2189 ; SIDE EFFECTS:
            ;2190 ;
            ;2191 ; d regsiter is destroyed
            ;2192 ;--
            ;2193 SET_TRAP_MASK:
U 118A, 0800,003C,0180,0A78,0000,118B  ;2194  d_r[0f]
U 118B, 0801,001C,0180,0AF8,0000,118C  ;2195  r[0f]_d.ornot.mask,d_alu ; Set mask
U 118C, 0001,003E,0180,0AF0,0000,0001  ;2196  r[0E]_d,return1    ; Save mask and return
            ;2197
            ;2198
            ;2199 ;+
            ;2200 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;2201 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;2202 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;2203 ;-
            ;2204
U 118D, 0010,0038,01C0,0978,0000,118E  ;2205 SETRCF: Q_RC[0F]
U 118E, 0019,2034,4DC0,0800,0000,118F  ;2206  Q_Q.AND.K[.FF00]
U 118F, 0019,2032,1D80,09F8,0000,0001  ;2207  RC[0F]_Q.OR.SC,RETURN1
U 12AA, 0000,003D,0180,0800,0000,116A  ;2208 12AA: ERROR1     ;U-ECO LOACTION
            ;2209
            ;2210
            ;2211
U 1190, 0818,0038,0180,0800,0000,105E  ;2212 DELAY: D_K[.8],J/CALLCON
            ;2213
            ;2214 ;
            ;2215 ; THIS ROUTINE IS USED TO INITIALIZE THE DW780. IT MUST BE CALLED WITH
            ;2216 ; THE DW780 BASE ADDRESS IN RC[0].
            ;2217 ;
            ;2218 INIT_DW780:
U 1191, 0810,0038,0180,0900,0000,1192  ;2219  D_RC[0]     ; GET BASE ADDRESS OF DW780
U 1192, 0019,0030,1180,0800,0200,1193  ;2220  VA_ALU,ALU_D.OR.K[.4]  ;
U 1193, 0818,0038,0580,0800,0000,1194  ;2221  D_K[.1]
U 1194, 0000,003E,0180,A800,0000,0001  ;2222  CACHE.P_D[LONG],RETURN1  ; INIT THE DW780
            ;2223
```

```
            ;2224 ;
            ;2225 ; THIS ROUTINE IS USED TO INITIALIZE THE CI780. IT MUST BE CALLED WITH
            ;2226 ; THE CI780 BASE ADDRESS IN RC[0].
            ;2227 ;
            ;2228 INIT_CI780:
U 1195. 0810.0038.0180.0900.0000.1196  ;2229  D_RC[0]     ; GET BASE ADDRESS OF CI780
U 1196. 0019.0030.1180.0800.0200.1197  ;2230  VA_ALU,ALU_D.OR.K[.4]  ; SET ADDRESS OF 'PMCSR' REG
U 1197. 0818.0038.0580.0800.0000.1198  ;2231  D_K[.1]
U 1198. 0018.0038.31C0.A800.0000.1199  ;2232  CACHE.P_D[LONG],Q_K[.40] ; INIT THE CI750
U 1199. 0819.2030.0980.0800.0000.119A  ;2233  D_Q.OR.K[.2]    ; 42(X)
U 119A. 0000.003E.0180.A800.0000.0001  ;2234  CACHE.P_D[LONG],RETURN1  ; SET 'PSA' AND 'MTD'
            ;2235 ;
            ;2236 ; THIS ROUTINE IS USED TO INITIALIZE THE RH780. IT MUST BE CALLED WITH
            ;2237 ; THE RH780 BASE ADDRESS IN RC[0].
            ;2238 ;
            ;2239 INIT_RH780:
U 119B. 0810.0038.0180.0900.0000.119C  ;2240  D_RC[0]     ; GET BASE ADDRESS OF RH780
U 119C. 0019.0030.1180.0800.0200.119D  ;2241  VA_ALU,ALU_D.OR.K[.4]  ;
U 119D. 0818.0038.0580.0800.0000.119E  ;2242  D_K[.1]
U 119E. 0000.003E.0180.A800.0000.0001  ;2243  CACHE.P_D[LONG],RETURN1  ; INIT THE RH780
            ;2244 ;*
            ;2245 ; THIS ROUTINE IS USED TO SETUP THE DW780 TO PERFORM A READ TO
            ;2246 ; PHYSICAL MEMORY LOCATION 0(X) AND PERFORM THE READ.
            ;2247 ;
            ;2248 ; ON ENTRY, RC[0] MUST CONTAIN ADDRESS OF CONFIGURATION REGISTER
            ;2249 ; AND R[7] MUST CONTAIN ADDRESS OF UNIBUS SPACE.
            ;2250 ;-
            ;2251 DW780_READ:
U 119F. 0810.0038.8580.0900.0084.71A0  ;2252  D_RC[0],SC_K[.C]  ; GENERATE ADDRESS OF MAP
U 11A0. 0000.003C.0580.0800.0084.B1A1  ;2253  SC_SC-K[.1]    ; REGISTER 0
            ;2254  VA_ALU,ALU_D.ORNOT.MASK, ; UBA BASE + 800
U 11A1. 0001.001C.8D80.0800.0284.71A2  ;2255       SC_K[.1F]
U 11A2. 0818.0038.7580.0800.0000.11A3  ;2256  D_K[.20]    ; GET PAGE NUMBER TO LOAD INTO MAP
U 11A3. 0801.001C.0180.0800.0000.11A4  ;2257  D_D.ORNOT.MASK    ; EQUALS 80000020
U 11A4. 0000.003C.0180.A800.0000.11A5  ;2258  CACHE.P_D[LONG]    ; LOAD THE MAP REGISTER
U 11A5. 0000.003C.8980.0800.0084.71A6  ;2259  SC_K[.D]    ; SETUP R[0F] TO CATCH TIMEOUT
U 11A6. 0003.001C.0180.0AF8.0000.11A7  ;2260  R[0F]_NOT.MASK    ; ...
U 11A7. 0810.0038.0180.0938.0000.11A8  ;2261  D_RC[7]     ; GET BASE OF UNIBUS SPACE
U 11A8. 0019.0030.7580.0800.0200.11A9  ;2262  VA_ALU,ALU_D.OR.K[.20]  ; LOAD BYTE ADDRESS WITHIN PAGE
U 11A9. 0000.403E.0180.C800.0000.0001  ;2263  D[WORD]_CACHE.P,RETURN1  ; EXECUTE THE READ, CAUSING THE ERROR
            ;2264
            ;2265 ;*
            ;2266 ; THIS ROUTINE IS USED TO CAUSE THE RH780 TO PERFORM A READ FROM
            ;2267 ; MEMORY. ON ENTRY, RC[0] MUST CONTAIN THE BASE ADDRESS OF THE RH780
            ;2268 ; AND R[8] MUST CONTAIN THE ADDRESS OF DRIVE 0 EXTERNAL REGISTER.
            ;2269 ;-
            ;2270 =0
            ;2271 RH780_READ:
U 104C. 0000.003D.0180.0800.0000.119B  ;2272  CALL[INIT_RH780]  ; FIRST INIT THE RH
U 104D. 0810.0038.85C0.0900.0084.71AA  ;2273  D_RC[0],SC_K[.C],Q_ALU  ; GENERATE ADDRESS OF MAP
U 11AA. 0000.003C.0580.0800.0084.B1AB  ;2274  SC_SC-K[.1]    ; REGISTER 0
            ;2275  VA_ALU,ALU_D.ORNOT.MASK, ; MBA BASE + 800
U 11AB. 0001.001C.8D80.0800.0284.71AC  ;2276       SC_K[.1F]
U 11AC. 0818.0038.7580.0800.0000.11AD  ;2277  D_K[.20]    ; GET PAGE NUMBER TO LOAD INTO MAP
U 11AD. 0801.001C.0180.0800.0000.11AE  ;2278  D_D.ORNOT.MASK    ; EQUALS 80000020
```

```
U 11AE, 0000,003C,0180,A800,0000,11AF  ;2279  CACHE.P_D[LONG]    ; LOAD THE MAP REGISTER
U 11AF, 0019,2030,6580,0800,0200,11B0  ;2280  VA_ALU,ALU_Q.OR.K[.10]  ; SET ADDRESS OF BYTE COUNT REG
U 11B0, 081B,0000,0980,0800,0000,11B1  ;2281  D_Q-K[.2]    ; GET BYTE COUNT
U 11B1, 0000,003C,0180,A800,0000,11B2  ;2282  CACHE.P_D[LONG]    ; LOAD THE BYTE COUNT REGISTER
U 11B2, 0019,2030,8580,0800,0200,11B3  ;2283  VA_ALU,ALU_Q.OR.K[.C]  ; SET ADDRESS OF VAR REGISTER
U 11B3, 0818,0038,7580,0800,0000,11B4  ;2284  D_K[.20]    ; GET VAR DATA
U 11B4, 0000,003C,0180,A800,0000,11B5  ;2285  CACHE.P_D[LONG]    ; LOAD THE VAR
U 11B5, 0019,2030,1180,0800,0200,11B6  ;2286  VA_ALU,ALU_Q.OR.K[.4]  ; SET ADDRESS OF CONTROL REGISTER
U 11B6, 0818,0038,0180,0800,0000,11B7  ;2287  D_K[.8]
U 11B7, 0000,003C,0180,A800,0000,11B8  ;2288  CACHE.P_D[LONG]    ; SET MAINTENANCE MODE
U 11B8, 0000,003C,0180,0A40,0200,11B9  ;2289  VA_R[8]     ; SET ADDRESS OF DRIVE 0 REGISTER
U 11B9, 0818,0038,7980,0800,0000,11BA  ;2290  D_K[.30]    ; GENERATE A WRITE TO DRIVE CMD
U 11BA, 0819,0030,0580,0800,0000,11BB  ;2291  D_D.OR.K[.1]    ; ...
U 11BB, 0000,003C,0180,A800,0000,1050  ;2292  CACHE.P_D[LONG]    ; START THE WRITE
            ;2293 =0 D_K[.10],
U 1050, 0818,0039,6580,0800,0000,1205  ;2294   CALL[WAIT.LOOP]
U 1051, 0019,2030,2180,0800,0200,11BC  ;2295  VA_ALU,ALU_Q.OR.K[.14]  ; SET ADDRESS OF DIAGNOSTIC REGISTER
U 11BC, 0000,003C,B980,0800,0084,71BD  ;2296  SC_K[.19]
U 11BD, 0803,001C,0180,0800,0000,11BE  ;2297  D_NOT.MASK
U 11BE, 0000,003C,0180,A800,0000,1052  ;2298  CACHE.P_D[LONG]    ; SET 'SIMULATE OCCUPIED'
            ;2299 =0 D_K[.10],
U 1052, 0818,0039,6580,0800,0000,1205  ;2300   CALL[WAIT.LOOP]
U 1053, 0000,003C,E580,0800,0084,71BF  ;2301  SC_K[.1A]
U 11BF, 0000,003C,0180,0800,0080,D1C0  ;2302  SC_SC+1
U 11C0, 0801,001C,0180,0800,0000,11C1  ;2303  D_D.ORNOT.MASK
U 11C1, 0000,003C,0180,A800,0000,1054  ;2304  CACHE.P_D[LONG]    ; SET 'S CLOCK' AND 'SIM_OCC'
            ;2305 =0 D_K[.10],
U 1054, 0818,0039,6580,0800,0000,1205  ;2306   CALL[WAIT.LOOP]
U 1055, 0801,0034,0180,0800,0000,11C2  ;2307  D_D.AND.MASK
U 11C2, 0000,003C,0180,A800,0000,1056  ;2308  CACHE.P_D[LONG]    ; CLEAR 'S CLOCK'
            ;2309 =0 D_K[.10],
U 1056, 0818,0039,6580,0800,0000,1205  ;2310   CALL[WAIT.LOOP]
U 1057, 0000,003C,E580,0800,0084,71C3  ;2311  SC_K[.1A]
U 11C3, 0801,001C,0180,0800,0000,11C4  ;2312  D_D.ORNOT.MASK
U 11C4, 0000,003C,0180,A800,0000,1058  ;2313  CACHE.P_D[LONG]    ; SET 'SIMULATE EBL' AND 'SIM_OCC'
            ;2314 =0 D_K[.10],
U 1058, 0818,0039,6580,0800,0000,1205  ;2315   CALL[WAIT.LOOP]
U 1059, 0801,0034,0180,0800,0000,11C5  ;2316  D_D.AND.MASK
U 11C5, 0000,003E,0180,A800,0000,0001  ;2317  CACHE.P_D[LONG],RETURN1   ; CLEAR 'SIM_EBL'
            ;2318
            ;2319 NOINTR: ;=================================================================
            ;2320   ;*
            ;2321   ;
            ;2322   ; UNEXPECTED INTERRUPT ERROR LOGOUT
            ;2323   ;
            ;2324   ;DATA: INTTERRUPT VECTOR STROBED
            ;2325   ; FAULT REGISTER  ID=^X1B
            ;2326   ; SBI ERROR REGISTER ID=^X19
            ;2327   ; TIMEOUT ADDRESS  ID=^X1A
            ;2328   ; MAINT REGISTER  ID=^X1D
            ;2329   ; CACHE PARITY ERROR REG ID=^X1E
            ;2330   ; TB ERROR REG 1  ID=^X13
            ;2331   ; TB ERROR REG 0  ID=^X12
            ;2332   ; TR LEVEL UNDER TEST
            ;2333   ; DEVICE FLAG (0-DW750, 1-RH750, 2-CI750, 3-DR750
```

```
            ;2334  ;-
            ;2335  ;=============================================================================
            ;2336  ;
U 11C6. 0000.003C.0180.0800.4000.11C7  ;2337  IEK/ISTR  ;STROBE THE VECTOR
U 11C7. 0000.003C.0180.0800.0000.11C8  ;2338  NOP
U 11C8. 0000.003C.35F0.2C00.0000.11C9  ;2339  Q_ID[VECTOR]
U 11C9. 0F19.2034.81C0.0800.0000.11CA  ;2340  Q_Q.AND.K[.3FF],D_0
U 11CA. 001D.0020.0180.0800.0010.11CB  ;2341  ALU_D[XOR]Q,CLK.UBCC
U 11CB. 0000.013C.0180.0800.0000.105C  ;2342  Z?
U 105C. 0000.003C.0180.0800.0000.11CC  ;2343 =0 J/ERRNON  ;NOT THE EXPECTED INTERRUPT
U 105D. 0000.003E.0180.0800.0000.0002  ;2344  RETURN2   ;YES THE CORRECT ONE OCCURED
U 11CC. 0810.0038.0180.0960.0000.11CD  ;2345 ERRNON: D_RC[0C]  ; GET DEVICE FLAG
U 11CD. 0001.003C.0180.09A8.0000.11CE  ;2346  RC[05]_D
U 11CE. 0001.203C.49F0.2DF0.0000.11CF  ;2347  RC[0E]_Q,Q_ID[TBER0]
U 11CF. 0001.203C.4DF0.2DB8.0000.11D0  ;2348  RC[07]_Q,Q_ID[TBER1]
U 11D0. 0001.203C.79F0.2DC0.0000.11D1  ;2349  RC[08]_Q,Q_ID[PARITY]
U 11D1. 0001.203C.75F0.2DC8.0000.11D2  ;2350  RC[09]_Q,Q_ID[MAINT]
U 11D2. 0001.203C.69F0.2DD0.0000.11D3  ;2351  RC[0A]_Q,Q_ID[TIME.ADDR]
U 11D3. 0001.203C.65F0.2DD8.0000.11D4  ;2352  RC[0B]_Q,Q_ID[SBI.ERR]
U 11D4. 0001.203C.6DF0.2DE0.0000.11D5  ;2353  RC[0C]_Q,Q_ID[FAULT]
U 11D5. 0001.203C.C5F0.2DE8.0000.1060  ;2354  RC[0D]_Q,Q_ID[T1]
U 1060. 0001.203D.F580.09B0.0084.718D  ;2355 =0 SETRCF[.A],RC[6]_Q  ;#ARG TO CONSOLE
U 1061. 0000.003E.0180.0800.0000.0001  ;2356  RETURN1
            ;2357 CLRSBI:
            ;2358 SBI.UNJAM:
            ;2359  ;=============================================================================
            ;2360  ;
            ;2361  ;PERFORM AN SBI UNJAM SEQUENCE
            ;2362  ;
            ;2363  ;=============================================================================
            ;2364  ;
U 11D6. 0F00.003C.0180.8000.0000.11D7  ;2365  D_0,MCT/SBI.HOLD  ;ASSERT HOLD FOR 16 CLOCK TICS
            ;2366 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,
U 11D7. 0019.0020.6180.8000.0010.11D8  ;2367   MCT/SBI.HOLD
U 11D8. 0000.013C.0180.8000.0000.1062  ;2368  Z?,MCT/SBI.HOLD
U 1062. 0819.0014.0580.8000.0000.11D7  ;2369 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM0
U 1063. 0F00.003C.0180.8800.0000.11D9  ;2370  D_0,MCT/SBI.HOLD+UNJAM  ;ASSERT HOLD AND UNJAM FOR 16
            ;2371       ; CLOCK TICS
            ;2372
            ;2373 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,
U 11D9. 0019.0020.6180.8800.0010.11DA  ;2374   MCT/SBI.HOLD+UNJAM
U 11DA. 0000.013C.0180.8800.0000.1064  ;2375  Z?,MCT/SBI.HOLD+UNJAM
U 1064. 0819.0014.0580.8800.0000.11D9  ;2376 =0 D_D+K[.1],MCT/SBI.HOLD+UNJAM,J/UNJAM1
U 1065. 0F00.003C.0180.8000.0000.11DB  ;2377  D_0,MCT/SBI.HOLD  ;FINISH SEQUENCE,ASSERT HOLD
            ;2378       ;  FOR THE ENDING 16 CLOCK TICS
            ;2379 UNJAM2: ALU_D[XOR]K[.F],CLK.UBCC,
U 11DB. 0019.0020.6180.8000.0010.11DC  ;2380   MCT/SBI.HOLD
U 11DC. 0000.013C.0180.8000.0000.1066  ;2381  Z?,MCT/SBI.HOLD
U 1066. 0819.0014.0580.8000.0000.11DB  ;2382 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM2
U 1067. 0000.003E.0180.0800.0000.0001  ;2383  RETURN1
            ;2384 INIT: ;=============================================================================
            ;2385  ;
            ;2386  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2387  ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2388  ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
```

```
            ;2389  ;
            ;2390  ;=====================================================================
            ;2391  ;
U 11DD, 0003,003C,0180,0AF8,0000,1068  ;2392  R[0F]_0     ;CLEAR U-TRAPS
U 1068, 0000,003D,0180,0800,0000,11D6  ;2393 =0 CALL[CLRSBI]    ; EXECUTE AN UNJAM SEQUENCE
U 1069, 0000,003C,0180,0800,0000,106A  ;2394  NOP
U 106A, 0000,003D,0180,0800,0000,11E0  ;2395 =0 CALL[SETPSL]    ;SET PSL AND CLEAR SOFTWARE
            ;2396        ;  INTRUPT REG
U 106B, 0000,003C,0180,0800,0000,106C  ;2397  NOP
U 106C, 0000,003D,0180,0800,0000,11E7  ;2398 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT BIT
U 106D, 0000,003C,0180,0800,0000,106E  ;2399  NOP
U 106E, 0000,003D,0180,0800,0000,11F1  ;2400 =0 CALL[CLRECC]    ;CLEAR ANY PENDING ECC INTRUPTS
U 106F, 0000,003C,0180,0800,0000,1070  ;2401  NOP
U 1070, 0000,003D,0180,0800,0000,11E3  ;2402 =0 CALL[CLRTIM]    ;CLEAR TIMEOUT BIT
U 1071, 0000,003C,0180,0800,0000,1072  ;2403  NOP
U 1072, 0000,003D,0180,0800,0000,11F5  ;2404 =0 CALL[ENBECC]    ;ENABLE ECC INTRUPTS
U 1073, 0000,003C,0180,0800,0000,1074  ;2405  NOP
U 1074, 0000,003D,0180,0800,0000,11EB  ;2406 =0 CALL[ENBFLT]    ;ENBALE FAULT INTERRUPTS
U 1075, 0818,0038,4580,0800,0000,1076  ;2407  D_K[.8000]    ;ENABLE CACHE PARITY ERROR
            ;2408 =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 1076, 0000,003D,7980,3C00,0000,116E  ;2409   ID[PARITY]_D
U 1077, 0F00,003C,0180,0800,0000,11DE  ;2410  D_0
U 11DE, 0000,003C,4980,3C00,0000,11DF  ;2411  ID[TBER0]_D    ;SHUT TB OFF
U 11DF, 0000,003E,7180,3C00,0000,0001  ;2412  ID[COMP]_D,RETURN1  ;DISABLE SILO INTERRUPTS
            ;2413 SETPSL: ;=====================================================================
            ;2414  ;
            ;2415  ;DROP THE CPU IPR LEVEL TO
            ;2416  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
            ;2417  ; PENDING
            ;2418  ;
            ;2419  ;=====================================================================
            ;2420  ;
U 11E0, 0F00,003C,0180,0800,0000,11E1  ;2421  D_0
U 11E1, 0000,003C,3980,3C00,0000,11E2  ;2422  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 11E2, 0000,003E,3D80,3C00,0000,0001  ;2423  RETURN1,ID[PSL]_D
            ;2424 CLRTIM: ;=====================================================================
            ;2425  ;
            ;2426  ;CLEAR THE CP TIMEOUT BIT IN THE
            ;2427  ; SBI ERROR REGISTER
            ;2428  ;
            ;2429  ;=====================================================================
            ;2430  ;
U 11E3, 0818,0038,0580,0800,0000,11E4  ;2431  D_K[.1]
U 11E4, 0000,003C,85F8,0800,0084,71E5  ;2432  Q_0,SC_K[.C]
U 11E5, 0D00,003C,0180,0800,0000,11E6  ;2433  D_DAL.SC
U 11E6, 0000,003E,6580,3C00,0000,0001  ;2434  ID[SBI.ERR]_D,RETURN1
            ;2435 CLRFLT: ;=====================================================================
            ;2436  ;
            ;2437  ;CLEAR THE CP FAULT BIT IN THE
            ;2438  ; FAULT REGISTER
            ;2439  ;
            ;2440  ;=====================================================================
            ;2441  ;
U 11E7, 0818,0038,0180,0800,0000,11E8  ;2442  D_K[.8]
U 11E8, 0000,003C,65F8,0800,0084,71E9  ;2443  Q_0,SC_K[.10]
```

```
U 11E9, 0D00,003C,0180,0800,0000,11EA  ;2444  D_DAL.SC
U 11EA, 0000,003E,6D80,3C00,0000,0001  ;2445  ID[FAULT]_D,RETURN1
            ;2446 ENBFLT: ;=================================================================
            ;2447  ;
            ;2448  ;ENABLE CPU FAULT INTERRUPT
            ;2449  ;
            ;2450  ;=================================================================
            ;2451  ;
U 11EB, 0818,0038,0580,0800,0000,11EC  ;2452  D_K[.1]
U 11EC, 0000,003C,D5F8,0800,0084,71ED  ;2453  Q_0,SC_K[.6]
U 11ED, 0D00,003C,0180,0800,0000,11EE  ;2454  D_DAL.SC
U 11EE, 0000,003C,8580,0800,0084,71EF  ;2455  SC_K[.C]
U 11EF, 0D00,003C,0180,0800,0000,11F0  ;2456  D_DAL.SC
U 11F0, 0000,003E,6D80,3C00,0000,0001  ;2457  ID[FAULT]_D,RETURN1
            ;2458 CLRECC: ;=================================================================
            ;2459  ;
            ;2460  ;CLEAR CRD,RDS BIT IN THE
            ;2461  ;SBI ERROR REGISTER
            ;2462  ;
            ;2463  ;=================================================================
            ;2464  ;
U 11F1, 0818,0038,0D80,0800,0000,11F2  ;2465  D_K[.3]
U 11F2, 0000,003C,89F8,0800,0084,71F3  ;2466  Q_0,SC_K[.D]
U 11F3, 0D00,003C,0180,0800,0000,11F4  ;2467  D_DAL.SC
U 11F4, 0000,003E,6580,3C00,0000,0001  ;2468  ID[SBI.ERR]_D,RETURN1
            ;2469 ENBECC: ;=================================================================
            ;2470  ;
            ;2471  ;ENABLE CRD,RDS INTERRUPTS
            ;2472  ;
            ;2473  ;=================================================================
            ;2474  ;
U 11F5, 0818,0038,0580,0800,0000,11F6  ;2475  D_K[.1]
U 11F6, 0000,003C,61F8,0800,0084,71F7  ;2476  Q_0,SC_K[.F]
U 11F7, 0D00,003C,0180,0800,0000,11F8  ;2477  D_DAL.SC
U 11F8, 0000,003E,6580,3C00,0000,0001  ;2478  ID[SBI.ERR]_D,RETURN1
            ;2479
            ;2480 ;
            ;2481 ; THIS ROUTINE IS USED TO WAIT FOR THE UNIBUS TO
            ;2482 ; COMPLETE ITS INITIALIZATION SEQUENCE
            ;2483 ;
            ;2484 WAITFORUNIBUS:
U 11F9, 0818,0038,C180,0800,0000,11FA  ;2485  D_K[.FFFF]
U 11FA, 0500,003C,0180,0800,0000,11FB  ;2486  D_D.LEFT
U 11FB, 0819,0000,0580,0800,0010,11FC  ;2487 LOOPUN: D_D-K[.1],CLK.UBCC
U 11FC, 0000,013C,0180,0800,0000,1078  ;2488  Z?
U 1078, 0000,003C,0180,0800,0000,11FB  ;2489 =0 J/LOOPUN
U 1079, 0000,003E,0180,0800,0000,0001  ;2490  RETURN1
            ;2491
```

```
            ;2492  .PAGE
            ;2493  .TOC "   Enable SBI"
            ;2494 ;++
            ;2495 ; FUNCTIONAL DESCRIPTION:
            ;2496 ;
            ;2497 ; This routine is used to ensure the SBI is enabled by clearing
            ;2498 ; bit<12> in the maintenance register.
            ;2499 ;
            ;2500 ; CALLING CONSTRAINT:
            ;2501 ;
            ;2502 ; =0
            ;2503 ;
            ;2504 ; SIDE EFFECTS:
            ;2505 ;
            ;2506 ; D & Q Registers destroyed
            ;2507 ;--
            ;2508 ENABLE.SBI:
U 11FD, 0000,003C,75F0,2C00,0000,11FE  ;2509  q_id[maint]  ; Get current value of maintenance reg
U 11FE, 0000,003C,8580,0800,0084,71FF  ;2510  sc_k[.c]  ; Setup to clear bit<12>
U 11FF, 0801,2034,0180,0800,0000,1200  ;2511  d_q.and.mask  ; Clear bit<12>
U 1200, 0000,003E,7580,3C00,0000,0001  ;2512  id[maint]_d,return1 ; and exit
```

```
            ;2513  .PAGE
            ;2514  .TOC "   Initialize the SBI"
            ;2515 ;**
            ;2516 ; FUNCTIONAL DESCRIPTION:
            ;2517 ;
            ;2518 ; This routine performs the following functions:
            ;2519 ;
            ;2520 ;  Executes an SBI Unjam
            ;2521 ;  Clears the SBI Fault Latch
            ;2522 ;  Clears the SBI Error Register
            ;2523 ;
            ;2524 ; CALLING CONSTRAINT:
            ;2525 ;
            ;2526 ; =0
            ;2527 ;
            ;2528 ; SIDE EFFECTS:
            ;2529 ;
            ;2530 ; D & Q Register destroyed
            ;2531 ;--
            ;2532 =0
            ;2533 SBI.INIT:
U 107A, 0000,003D,0180,0800,0000,11D6  ;2534  call[sbi.unjam]  ; Unjam the SBI
U 107B, 0818,0038,C180,0800,0000,1201  ;2535  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1201, 0801,0028,6580,3C00,0000,1202  ;2536  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1202, 0F00,003C,6D80,3C00,0000,1203  ;2537  id[fault]_d,d_0
U 1203, 0000,003C,6D80,3C00,0000,1204  ;2538  id[fault]_d
U 1204, 0000,003E,6580,3C00,0000,0001  ;2539  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2540  .PAGE
            ;2541  .TOC "   Wait Loop Routine"
            ;2542 ;**
            ;2543 ; FUNCTIONAL DESCRIPTION:
            ;2544 ;
            ;2545 ; This routine is used to wait the specified number of machine cycles.
            ;2546 ; This routine is typically called to wait for an SBI write to
            ;2547 ; I/O space to complete.
            ;2548 ;
            ;2549 ; CALLING CONSTRAINT:
            ;2550 ;
            ;2551 ; =0
            ;2552 ;
            ;2553 ; INPUTS:
            ;2554 ;
            ;2555 ; d - Contains number of cycles to wait
            ;2556 ; alu.z condition code must not be set
            ;2557 ;
            ;2558 ; SIDE EFFECTS:
            ;2559 ;
            ;2560 ; d is destroyed
            ;2561 ;--
            ;2562 WAIT.LOOP:
U 1205, 0018,0038,6180,0800,0010,107C  ;2563  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2564      ; ...is not set
            ;2565 =0
            ;2566 W.LOOP:
            ;2567  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 107C, 0819,0100,0580,0800,0010,107C  ;2568       j/W.LOOP
U 107D, 0000,003E,0180,0800,0000,0001  ;2569  return1    ; exit
            ;2570
            ;2571
```

```
            ;2572  .PAGE
            ;2573  .TOC "   Find MS780-E Memory"
            ;2574 ;++
            ;2575 ; FUNCTIONAL DESCRIPTION:
            ;2576 ;
            ;2577 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2578 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2579 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2580 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2581 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2582 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2583 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2584 ;     (zero).
            ;2585 ;     All the other MS780-E/H's found will have their starting  address
            ;2586 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2587 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2588 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2589 ;     current MS780-E unit under test.
            ;2590 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2591 ;     set to maximum.
            ;2592 ;
            ;2593 ; CALLING CONSTRAINT:
            ;2594 ;
            ;2595 ; =00
            ;2596 ;
            ;2597 ; OUTPUTS:
            ;2598 ;
            ;2599 ; rc[0e] - Contains address of Configuration Register
            ;2600 ; r[0]  - Contains TR level
            ;2601 ;
            ;2602 ; SIDE EFFECTS:
            ;2603 ;
            ;2604 ; D register is destroyed.
            ;2605 ;--
            ;2606 =0
            ;2607 FIND.MS780E:
U 107E, 0000,003D,0180,0800,0000,107A  ;2608  call[sbi.init]  ; Make sure SBI is enabled
U 107F, 0003,003C,0180,0988,0000,1206  ;2609  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1206, 0818,0038,1980,0800,0000,1207  ;2610  d_k[zero]  ; Clear MS780-E/H counter
U 1207, 0000,003C,F980,3C00,0000,1208  ;2611  id[3e]_d  ; ...
U 1208, 0018,0038,0580,0A80,0000,1209  ;2612  r[0]_k[.1]  ; Init TR counter
U 1209, 0F03,003C,0180,09F0,0000,1080  ;2613  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2614      ; ...CNFG A Reg
            ;2615 =0
            ;2616 FIND.MEM.LOOP:
U 1080, 0800,003D,0180,0A00,0000,122F  ;2617  d_r[0],call[generate.io]; Generate address of TR level
U 1081, 0001,003C,0180,09F0,0200,1082  ;2618  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1082, 0000,003D,8980,0800,0084,718A  ;2619 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2620  d[long]_cache.p, ; Read the specified tr level
U 1083, 0000,003C,0180,C970,0000,120A  ;2621   lc_rc[0e]  ; ...
U 120A, 0000,003C,0180,0A78,0010,120B  ;2622  alu_r[0f],clk.ubcc ; Check for no response
U 120B, 0001,013C,0180,0880,0082,1084  ;2623  z?,sc_d,la_ra[0] ; Branch if no adapter here
U 1084, 0000,003C,0180,0800,0000,120C  ;2624 =0 j/check.adpt.code ; Check for a memory
U 1085, 0000,003C,0180,0800,0000,122B  ;2625  j/find.mem.lp1  ; Try next tr
            ;2626 CHECK.ADPT.CODE:
```

```
U 120C, 0000,003C,1D80,0800,1404,720D  ;2627  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 120D, 0000,163C,0180,0800,0000,1008  ;2628  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,720E  ;2629 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,720F  ;2630  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,122B  ;2631  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,122B  ;2632  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7214  ;2633  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,122B  ;2634  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,1218  ;2635  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,1218  ;2636  rc[0e]_lc,j/found.ms780e ; MS780-E
            ;2637 ;
            ;2638 ; Found an MS780-A or -C. Set the starting address
            ;2639 ; to maximum.
            ;2640 ;
U 120E, 0818,0038,5D80,0800,0000,1210  ;2641 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 120F, 0818,0038,0580,0800,0000,1210  ;2642 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
            ;2643 MS780.COMMON:
U 1210, 0819,0030,7180,0800,0000,1211  ;2644  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1211, 0000,003C,01F8,0803,0080,D212  ;2645  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1212, 0D00,003C,0180,0800,0000,1213  ;2646  d_dal.sc  ; Put data in bits<29:14>
            ;2647  cache.p_d[long], ; Set the starting address to maximum
U 1213, 0000,003C,0180,A800,0000,122B  ;2648       j/find.mem.lp1 ; and continue TR scen
            ;2649 ;
            ;2650 ; Found an MA780. Set the starting address to maximum
            ;2651 ;
            ;2652 MA780.MAX:
U 1214, 0818,0038,39F8,0803,0000,1215  ;2653  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1215, 0D00,003C,0180,0803,0000,1216  ;2654  d_dal.sc,va_va+4 ; Put in proper position
U 1216, 0000,003C,0180,0803,0000,1217  ;2655  va_va+4   ; Point to Port Invalidate Ctrl Register
            ;2656  cache.p_d[long], ; Set starting address to maximum
U 1217, 0000,003C,0180,A800,0000,122B  ;2657       j/find.mem.lp1
            ;2658 ;
            ;2659 ; Found an MS780E
            ;2660 ;
            ;2661 FOUND.MS780E:
U 1218, 0000,003C,F9F0,2C00,0000,1219  ;2662  q_id[3e]  ; Set up to see how many MS780-E's
            ;2663  alu_q.xor.k[.3], ; Are there Maximum units?
U 1219, 0019,2020,0D80,0800,0010,121A  ;2664   clk.ubcc ; Check condition codes
U 121A, 0000,013C,0180,0800,0000,1086  ;2665  z?    ; Are we at maximum units for system
U 1086, 0000,003C,0180,0800,0000,121B  ;2666 =0 J/ms780e.tst  ; No go find how many units ther are
U 1087, 0818,0038,C180,0800,0000,1088  ;2667  d_k[.ffff]  ; Yes set address to maximum
U 1088, 0000,003D,0180,0800,0000,1233  ;2668 =0 call[set.start.addr] ; .....
U 1089, 0000,003C,0180,0800,0000,122B  ;2669  j/find.mem.lp1  ; Go check to see if we are done
            ;2670 ;
            ;2671 MS780E.TST:
            ;2672  alu_rc[01],  ; Check "Found MS780E Flag"
U 121B, 0010,0038,0180,0908,0010,121C  ;2673   clk.ubcc ; Check condition codes
U 121C, 0810,0138,0180,0970,0000,108A  ;2674  z?,d_rc[0e]  ; Is this unit the first MS780E ?
            ;2675 =0 j/not.first.ms780e, ; No
U 108A, 0818,0038,C180,0800,0000,108E  ;2676   d_k[.ffff] ; Set starting address
U 108B, 0001,003C,0180,0988,0000,121D  ;2677  rc[01]_d  ; Yes put base address in rc[01]
U 121D, 0818,0038,7980,0800,0000,121E  ;2678  d_k[.30]  ; Set up SC to point to first unit
U 121E, 0019,0014,F580,0800,0082,121F  ;2679  alu_d+k[.a],sc_alu ; ...
U 121F, 0810,0038,0180,0908,0000,1220  ;2680  d_rc[01]  ; Put base address of unit in D
U 1220, 0000,003C,0180,3400,0000,1221  ;2681  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1221, 0F00,003C,0180,0800,0000,108C  ;2682  d_0   ; Prepare to set starting address to Zero
U 108C, 0000,003D,0180,0800,0000,1233  ;2683 =0 call[set.start.addr] ; Go do it
            ;2684  j/find.mem.lp1,  ; Check to see if we are done
U 108D, 0003,003C,0180,09E8,0000,122B  ;2685   rc[0d]_0 ; ....
            ;2686 =0
            ;2687 NOT.FIRST.MS780E:
U 108E, 0000,003D,0180,0800,0000,1233  ;2688  call[set.start.addr] ; Go set starting address to maximum
U 108F, 0000,003C,F9F0,2C00,0000,1222  ;2689  q_id[3e]  ; Get the number of MS780-Es found
U 1222, 0819,2014,0580,0800,0000,1223  ;2690  d_q+k[.1]  ; Increment this counter
U 1223, 0000,003C,F980,3C00,0000,1224  ;2691  id[3e]_d  ; Save the counter
U 1224, 0018,0038,11C0,0800,0000,1225  ;2692  q_k[.4]   ; Prepare to form pointer to the ID registers
            ;2693     ; ...were the I/O base addresses will be stored
U 1225, 081D,2000,7980,0800,0000,1226  ;2694  d_q-d,k[.30]  ; ... adjust pointer
U 1226, 0819,0014,7980,0800,0000,1227  ;2695  d_d+k[.30]  ; Point the SC to the ID register
U 1227, 0000,003C,F580,0800,0000,1228  ;2696  k[.a]   ; ...to receive I/O base address
U 1228, 0019,0014,F580,0800,0082,1229  ;2697  alu_d+k[.a],sc_alu ; ...
U 1229, 0810,0038,0180,0970,0000,122A  ;2698  d_rc[0e]  ; Get base address to d
U 122A, 0000,003C,0180,3400,0000,122B  ;2699  id(sc)_d  ; Move base address to ID pointed to by SC
            ;2700
            ;2701 ;
            ;2702 ; Try next tr
            ;2703 ;
            ;2704 FIND.MEM.LP1:
U 122B, 0818,0014,0580,0A80,0000,122C  ;2705  r[0]_ia+k[.1],d_alu ; Increment TR level
            ;2706  alu_d.and.k[.f],
U 122C, 0019,0034,6180,0800,0010,122D  ;2707      clk.ubcc  ; Check if all done
U 122D, 0000,013C,0180,0800,0000,1090  ;2708  z?   ; Branch if yes
U 1090, 0000,003C,0180,0800,0000,1080  ;2709 =0 j/find.mem.loop  ; Try next TR
U 1091, 0810,0038,0180,0908,0010,122E  ;2710  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 122E, 0000,013C,0180,0800,0000,1092  ;2711  z?   ; Check condition codes
U 1092, 0001,003E,0180,09F0,0000,0002  ;2712 =0 return2,rc[0e]_d ; Yes MS780E was found
U 1093, 0000,003E,0180,0800,0000,0001  ;2713  return1   ; No MS780E found
            ;2714
```

```
            ;2715  .PAGE
            ;2716  .TOC "   Generate IO Address"
            ;2717 ;++
            ;2718 ; FUNCTIONAL DESCRIPTION:
            ;2719 ;
            ;2720 ; This routine is used to generate an SBI Configuration Register
            ;2721 ; address from a TR level.
            ;2722 ;
            ;2723 ; CALLING CONSTRAINT:
            ;2724 ;
            ;2725 ; =0
            ;2726 ;
            ;2727 ; INPUTS:
            ;2728 ;
            ;2729 ; D - Contains TR level
            ;2730 ;
            ;2731 ; OUTPUTS:
            ;2732 ;
            ;2733 ; D - Contains SBI IO address
            ;2734 ;--
            ;2735 GENERATE.IO:
U 122F, 0000,003C,89F8,0800,0084,7230  ;2736  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1230, 0D00,003C,8D80,0800,0084,7231  ;2737  d_dal.sc,sc_k[.1f] , Put TR level in place, setup to
U 1231, 0000,003C,0980,0800,0084,B232  ;2738  sc_sc-k[.2]  ; insert bit 29
U 1232, 0801,001E,0180,0800,0000,0001  ;2739  d_d.ornot.mask,return1 ; and return
            ;2740
            ;2741
```

```
            ;2742  .PAGE
            ;2743  .TOC "    Set Starting Address"
            ;2744 ;++
            ;2745 ; FUNCTIONAL DESCRIPTION:
            ;2746 ;
            ;2747 ; This routine is used to set the starting address of the
            ;2748 ; memory to the specified value.
            ;2749 ;
            ;2750 ; CALLING CONSTRAINT:
            ;2751 ;
            ;2752 ; =0
            ;2753 ;
            ;2754 ; INPUTS:
            ;2755 ;
            ;2756 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2757 ;     (B Register Image divided by 2**16)
            ;2758 ; rc[0] - Contains Address of Register A
            ;2759 ;
            ;2760 ; OUTPUTS:
            ;2761 ;
            ;2762 ; d - Contains aligned starting address (B Register Image)
            ;2763 ;
            ;2764 ; SIDE EFFECTS:
            ;2765 ;
            ;2766 ; d,q,va, and sc are destroyed
            ;2767 ;--
            ;2768 SET.START.ADDR:
U 1233, 0010,0038,6580,0970,0284,7234  ;2769  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1234, 0000,003C,01F8,0803,0000,1235  ;2770  va_va+4,q_0  ; Set address to Register B
            ;2771  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1235, 0D00,003C,0980,0800,0084,B236  ;2772      sc_sc-k[.2]  ; Setup to insert bit 14
U 1236, 0801,001C,0180,0800,0000,1237  ;2773  d_d.ornot.mask  ; Insert Write Enable bit
U 1237, 0000,003E,0180,A800,0000,0001  ;2774  cache.p_d[long],return1 ; Set the starting address and return
            ;2775
            ;2776
            ;2777
```

```
            ;2778  .PAGE
            ;2779  .TOC "   ENABLE NEXT MS780-E"
            ;2780 ;++
            ;2781 ; FUNCTIONAL DESCRIPTION:
            ;2782 ;
            ;2783 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2784 ;    They will be tested  separately, according to the TR level at which
            ;2785 ;    they are located,  starting with the one  at the lowest level first.
            ;2786 ;          When  a test has finished  with the first unit, it will have to call
            ;2787 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2788 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2789 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2790 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2791 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2792 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2793 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2794 ;    found - 1 in register 3E. In this subroutine the SC register is used
            ;2795 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2796 ;
            ;2797 ; CALLING CONSTRAINT:
            ;2798 ;
            ;2799 ;  =DD
            ;2800 ;
            ;2801 ;--
            ;2802 ENABLE.NEXT.MS780E:
U 1238, 0000,003C,F9F0,2C00,0000,1239  ;2803  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2804  ALU_Q,   ; Check to see if it is zero
U 1239, 0001,203C,0180,0800,0010,123A  ;2805     CLK.UBCC  ; Check Condition Codes
U 123A, 0000,013C,0180,0800,0000,1094  ;2806  Z?   ; ...for zero
U 1094, 0000,003C,0180,0800,0000,123B  ;2807 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 1095, 0000,003E,0180,0800,0000,0002  ;2808  RETURN2   ; Zero No more units to test
            ;2809 ENABLE.NEXT:
U 123B, 0818,0038,C180,0800,0000,1096  ;2810  D_K[.FFFF]  ; Load D with Maximum Starting address
U 1096, 0000,003D,0180,0800,0000,1233  ;2811 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 1097, 0818,0038,7980,0800,0000,123C  ;2812  D_K[.30]  ; Prepare to point to the ID register holding
            ;2813     ; ...the I/O Base address of the next MS780-E
U 123C, 0819,0014,1180,0800,0000,123D  ;2814  D_D+K[.4]  ; ...
U 123D, 0000,003C,F580,0800,0000,123E  ;2815  K[.A]   ; ...
U 123E, 0819,0014,F580,0800,0000,123F  ;2816  D_D+K[.A]  ; ...
U 123F, 0000,003C,F9F0,2C00,0000,1240  ;2817  Q_ID[3E]  ; Get MS780-E Counter
            ;2818  D_D-Q,   ; D now holds the pointer to the ID register
U 1240, 081D,0000,0180,0800,0082,1241  ;2819    SC_ALU  ; ...
U 1241, 0000,003C,01F0,2400,0000,1242  ;2820  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1242, 0001,203C,0180,09F0,0000,1243  ;2821  RC[0E]_Q  ; Save it also in RC[0E]
U 1243, 0F03,003C,0180,09E8,0000,1098  ;2822  RC[0D]_0,D_0  ; Set starting address to zero
U 1098, 0000,003D,0180,0800,0000,1233  ;2823 =0 CALL[SET.START.ADDR] ; ....
U 1099, 0F00,003C,F9F0,2C00,0000,1244  ;2824  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 1244, 081D,2008,0180,0800,0000,1245  ;2825  D_Q-D-1   ; Subtract 1 from total
U 1245, 0000,003C,F980,3C00,0000,109A  ;2826  ID[3E]_D  ; Adjust the pointer
U 109A, 0000,003D,0180,0800,0000,1189  ;2827 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 109B, 0000,003E,0180,0800,0000,0001  ;2828  RETURN1   ; Tell caller another unit was found
            ;2829
```

```
            ;2830  .PAGE
            ;2831  .TOC "   CONFIGURE MS780-E/H"
            ;2832 ;*************************************************************
            ;2833 ;++
            ;2834 ;
            ;2835 ; Functional Description:
            ;2836 ; ----------------------
            ;2837 ;
            ;2838 ;       This  subroutine  will  be  used  by  tests  that   do   not
            ;2839 ; specifically  check  the  memory, but  make use of it to execute.
            ;2840 ; Its purpose is to find a MS780-E and  configure  it  properly  so
            ;2841 ; that  if  a  Read/Write operation will take place no false errors
            ;2842 ; will be logged due to misconfigured memory.
            ;2843 ;
            ;2844 ; Calling Constraint: =00
            ;2845 ; ------------------
            ;2846 ;
            ;2847 ;
            ;2848 ; Inputs:
            ;2849 ; ------
            ;2850 ;
            ;2851 ; RC[0E] - I/O BASE OF MS780-E/H
            ;2852 ;
            ;2853 ;
            ;2854 ; Outputs:
            ;2855 ; -------
            ;2856 ;
            ;2857 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
            ;2858 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
            ;2859 ;       OR IF NO MS780-E/Hs WERE FOUND
            ;2860 ;       ON THE SBI
            ;2861 ;
            ;2862 ; Side Effects:
            ;2863 ; ------------
            ;2864 ;
            ;2865 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
            ;2866 ;
            ;2867 ;
            ;2868 ;
            ;2869 ;--
            ;2870 ;*************************************************************
            ;2871 CONFIG.MS780E:
U 1246,  0818,0038,0D80,0800,0000,1247  ;2872  D_K[.3]     ; Set up flag for the Fail Chain
U 1247,  0001,003C,0180,09E8,0000,1010  ;2873  RC[0D]_D     ; ...
U 1010,  0000,003D,0180,0800,0000,107E  ;2874 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1011,  0000,003D,0980,0800,0084,7048  ;2875  ERROR1[.2]    ; No MS780-E/H's found on the SBI
U 1012,  0000,003C,0180,0800,0000,1018  ;2876  NOP     ; OK. Found at least one MS780-E/H
            ;2877 =
            ;2878 CONFIG.RETRY:     ; Entry point for the case in which the
            ;2879       ; ...first MS780-E/H could not be
            ;2880       ; ...properly configured
U 1018,  0000,003D,0180,0800,0000,1038  ;2881 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1019,  0000,003C,0180,0800,0000,1030  ;2882  J/CNFG.LMIS    ; Lower controller misconfigured
U 101A,  0000,003E,0180,0800,0000,0001  ;2883  RETURN1     ; OK. Lower Controller is configured
            ;2884 =
```

```
            ;2885 =00
            ;2886 CNFG.LMIS:
U 1030, 0000,003D,0180,0800,0000,103C  ;2887  CALL[SELECT.UPPER]  ; Select the upper controller
U 1031, 0000,003C,0180,0800,0000,1034  ;2888  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 1032, 0000,003E,0180,0800,0000,0001  ;2889  RETURN1    ; OK. Upper Controller is configured
            ;2890 =
            ;2891 =00
            ;2892 CNFG.UMIS:
U 1034, 0000,003D,0180,0800,0000,1238  ;2893  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2894       ; ...MS780-E/H found so go and see
            ;2895       ; ...if there are any more
U 1035, 0000,003C,0180,0800,0000,1248  ;2896  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2897       ; ...try to configure it
U 1036, 0818,0038,0D80,0800,0000,1037  ;2898  D_K[.3]    ; Set Flag for Fail Chain
U 1037, 0001,003C,0180,09E8,0000,1248  ;2899  RC[0D]_D    ; ...
U 1248, 0000,003D,0980,0800,0084,7048  ;2900  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2901       ; ...abort the test
            ;2902 =
            ;2903
```

```
                ;2904  .PAGE
                ;2905  .TOC "    MS780-E/H CNFG REG CLEANUP"
                ;2906 ;*************************************************************
                ;2907 ;**
                ;2908 ;
                ;2909 ; Functional Description:
                ;2910 ; ----------------------
                ;2911 ;
                ;2912 ;      This subroutine is used to clear all error conditions
                ;2913 ; in the MS780-E/H Configuration/Status/Error Registers.
                ;2914 ; Bits beeing cleared are:
                ;2915 ;
                ;2916 ; CNFG Regsiter B <11:00>
                ;2917 ;
                ;2918 ; CNFG Register C and D <31:28> and <10:06>
                ;2919 ;   ---
                ;2920 ;
                ;2921 ;
                ;2922 ; Calling Constraint: =0
                ;2923 ; ------------------
                ;2924 ;
                ;2925 ;
                ;2926 ; Inputs:
                ;2927 ; ------
                ;2928 ;
                ;2929 ; RC[0E] MUST contain the I/O base address of
                ;2930 ; the MS780-E/H currently under test
                ;2931 ;
                ;2932 ; Outputs:
                ;2933 ; -------
                ;2934 ;
                ;2935 ; NO specific outputs.
                ;2936 ; Above mentioned bits will be cleared.
                ;2937 ;
                ;2938 ; Side Effects:
                ;2939 ; ------------
                ;2940 ;
                ;2941 ; The contents of the following registers will be lost
                ;2942 ; Q, D, SC, VA
                ;2943 ;
                ;2944 ;
                ;2945 ;--
                ;2946 ;*************************************************************
                ;2947 MS780E.CLEANUP:
U 1249, 0010,0038,0180,0918,0200,124A  ;2948  VA_RC[3]    ; Point to CNFG Reg A of MS780-E/H
U 124A, 0000,003C,0180,C800,0000,124B  ;2949  D[LONG]_CACHE.P    ; Clear SBI Faults
                ;2950  D_0,     ; Get data to clear Read/Write bits
U 124B, 0F00,003C,0180,0803,0000,124C  ;2951   VA_VA+4    ; Point to CNFG Reg B
U 124C, 0000,003C,0180,A800,0000,124D  ;2952  CACHE.P_D[LONG]    ; Clear Read/Write bits in CNFG reg B
U 124D, 0818,0038,7980,0800,0000,124E  ;2953  D_K[.30]    ; Prepare to form x30000780 to clear
                ;2954          ; ...CNFG Reg's C and D
U 124E, 0B00,003C,0180,0800,0000,124F  ;2955  D_D.SWAP    ; D = x30000000
                ;2956  Q_D,     ; Save in the Q Reg
U 124F, 0818,0038,CDE0,0800,0000,1250  ;2957   D_K[.F0]   ; D = xF0
U 1250, 0000,003C,0D80,0800,0084,7251  ;2958  SC_K[.3]    ; Must shift D Reg left 3 bits to get
```

```
            ;2959       ; ...x780
U 1251, 0D00,003C,0180,0800,0000,1252  ;2960  D_DAL.SC    ; D = x780
            ;2961  D_D.OR.Q,    ; D = x30000780
U 1252, 081D,0030,0180,0803,0000,1253  ;2962   VA_VA+4    ; Point to CNFG Reg C
U 1253, 0000,003C,0180,A800,0000,1254  ;2963  CACHE.P_D[LONG]   ; Clear Bits in CNFG Reg C
U 1254, 0000,003C,0180,0800,0000,1255  ;2964  NOP
U 1255, 0000,003C,0180,A800,0000,1256  ;2965  CACHE.P_D[LONG]   ; Do second write to make sure that
            ;2966       ; ...uSequencer Parity error bit
            ;2967       ; ...is clear
U 1256, 0000,003C,0180,0803,0000,1257  ;2968  VA_VA+4    ; Point to CNFG Reg D
U 1257, 0000,003C,0180,A800,0000,1258  ;2969  CACHE.P_D[LONG]   ; Clear bits in CNFG Reg D
U 1258, 0000,003C,0180,0800,0000,1259  ;2970  NOP
            ;2971  CACHE.P_D[LONG],  ; Do second write to clear uSequencer
            ;2972       ; ...Parity error bit
U 1259, 0000,003E,0180,A800,0000,0001  ;2973   RETURN1   ; Return to caller
            ;2974
            ;2975
```

```
            ;2976  .PAGE
            ;2977  .TOC "   Select Controller"
            ;2978 ;++
            ;2979 ; FUNCTIONAL DESCRIPTION:
            ;2980 ;
            ;2981 ; This routine is used to put the memory system into the
            ;2982 ; specified configuration with respect to controller select.
            ;2983 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2984 ; a RETURN2 is made.
            ;2985 ;
            ;2986 ; CALLING CONSTRAINT:
            ;2987 ;
            ;2988 ; =00
            ;2989 ;
            ;2990 ; INPUTS:
            ;2991 ;
            ;2992 ; d - Contains value to write to bits<2:0> of Register A
            ;2993 ; rc[0e] - Address of Register A
            ;2994 ;
            ;2995 ; SIDE EFFECTS:
            ;2996 ;
            ;2997 ; sc,d,va are destroyed
            ;2998 ;--
            ;2999 SELECT.CNTRLR:
U 125A, 0010.0038.0180.0970.0284.725B  ;3000  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 125B, 0801.001C.0180.0800.0000.125C  ;3001  d_d.ornot.mask  ; Insert the enable bit into D reg
U 125C, 0000.003C.0180.A800.0000.125D  ;3002  cache.p_d[long]  ; Write the configuration Register
U 125D, 0818.0038.5D80.0800.0000.125E  ;3003  d_k[.7]   ; Get Misconfiguration bits
U 125E, 0000.003C.61F8.0800.0084.725F  ;3004  sc_k[.F],q_0  ; ...
U 125F, 0D00.003C.0180.0800.0000.1260  ;3005  d_del.sc  ; ... D = x38000
U 1260, 0000.003C.01E0.0800.0000.1261  ;3006  q_d   ; Save mask
U 1261, 0000.003C.0180.C800.0000.1262  ;3007  d[long]_cache.p  ; Read CNFG A Reg and
            ;3008  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1262, 001D.2034.0180.0800.0010.1263  ;3009   clk.ubcc ; ...
U 1263, 0000.013C.0180.0800.0000.109C  ;3010  z?   ; Branch if no
U 109C, 0000.003E.0180.0800.0000.0001  ;3011 =0 RETURN1   ; Bad configuration
U 109D, 0000.003E.0180.0800.0000.0002  ;3012  RETURN2   ; Configuration ok
            ;3013
            ;3014
```

```
;3015  .PAGE
;3016  .TOC "   SELECT LOWER CONTROLLER"
;3017 ;*************************************************************
;3018 ;++
;3019 ;
;3020 ; Functional Description:
;3021 ; ----------------------
;3022 ;
;3023 ; This subroutine can be called to select the lower
;3024 ; Controller on a MS780-E/H.
;3025 ; If the Lower controller is misconfigured then a
;3026 ; RETURN1 will be made.  Otherwise a RETURN2
;3027 ;
;3028 ; Calling Constraint: =00
;3029 ; ------------------
;3030 ;
;3031 ;
;3032 ; Inputs:
;3033 ; ------
;3034 ;
;3035 ; RC[0E] Must hold the I/O base address of the MS780-E/H
;3036 ;
;3037 ; Outputs:
;3038 ; -------
;3039 ;
;3040 ; RC[0D] will have the address of CNFG Register C
;3041 ;       ( 2000x008 )
;3042 ;
;3043 ; Side Effects:
;3044 ; ------------
;3045 ;
;3046 ; Contents of the following registers will lost:
;3047 ;
;3048 ; D
;3049 ;
;3050 ; The Lower controller on the MS780-E/H will be configured
;3051 ; when the subroutine is exitted with a RETURN2
;3052 ;--
;3053 ;*************************************************************
;3054 =00
;3055 SELECT.LOWER:
;3056  D_K[ZERO],    ; Interleave mode value for CNFG Reg A
U 1038, 0818,0039,1980,0800,0000,125A  ;3057   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1039, 0000,003E,0180,0800,0000,0001  ;3058  RETURN1     ; Lower Controller Misconfigured
U 103A, 0810,0038,0180,0970,0000,103B  ;3059  D_RC[0E]    ; Get MS780-E/H I/O Base address
;3060  RC[0D] D+K[.8],    ; Set the address of CNFG Reg D
U 103B, 0019,0016,0180,09E8,0000,0002  ;3061   RETURN2    ; Let caller know that the controller
;3062       ; ...was configured properly
;3063
```

```
            ;3064  .PAGE
            ;3065  .TOC "   SELECT UPPER CONTROLLER"
            ;3066 ;*****************************************************
            ;3067 ;++
            ;3068 ;
            ;3069 ; Functional Description:
            ;3070 ; ----------------------
            ;3071 ;
            ;3072 ; This subroutine can be called to select the upper
            ;3073 ; Controller on a MS780-E/H.
            ;3074 ; If the Upper controller is misconfigured then a
            ;3075 ; RETURN1 will be made.  Otherwise a RETURN2
            ;3076 ;
            ;3077 ; Calling Constraint: =00
            ;3078 ; ------------------
            ;3079 ;
            ;3080 ;
            ;3081 ; Inputs:
            ;3082 ; ------
            ;3083 ;
            ;3084 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;3085 ;
            ;3086 ; Outputs:
            ;3087 ; -------
            ;3088 ;
            ;3089 ; RC[0D] will have the address of CNFG Register D
            ;3090 ;        ( 2000x010 )
            ;3091 ;
            ;3092 ; Side Effects:
            ;3093 ; ------------
            ;3094 ;
            ;3095 ; Contents of the following registers will lost:
            ;3096 ;
            ;3097 ; D
            ;3098 ;
            ;3099 ; The Upper controller on the MS780-E/H will be configured
            ;3100 ; when the subroutine is exitted with a RETURN2
            ;3101 ;--
            ;3102 ;*****************************************************
            ;3103 =00
            ;3104 SELECT.UPPER:
            ;3105   D_K[.2],    ; Interleave mode value for CNFG Reg A
U 103C, 0818,0039,0980,0800,0000,125A  ;3106    CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 103D, 0000,003E,0180,0800,0000,0001  ;3107  RETURN1     ; Upper Controller Misconfigured
U 103E, 0810,0038,0180,0970,0000,103F  ;3108  D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;3109   RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 103F, 0019,0016,8580,09E8,0000,0002  ;3110    RETURN2    ; Let caller know that the controller
            ;3111        ; ...was configured properly
            ;3112
```

```
            ;3113  .PAGE
            ;3114  .TOC "   SET UP MS780-E RDS"
            ;3115 ;*****************************************************************
            ;3116 ;**
            ;3117 ;
            ;3118 ; Functional Description:
            ;3119 ; ----------------------
            ;3120 ;
            ;3121 ; This subtest sets up the memory so that a RDS will
            ;3122 ; take place when location 4020 is read.
            ;3123 ; The MS780-E is placed in Diagnostic mode 2 (ECC Bypass)
            ;3124 ; and locations 4020 and 4024 are intialized to x03 and
            ;3125 ; 0 respectively.
            ;3126 ;
            ;3127 ;
            ;3128 ; Calling Constraint: =0
            ;3129 ; ------------------
            ;3130 ;
            ;3131 ;
            ;3132 ; Inputs:
            ;3133 ; ------
            ;3134 ;
            ;3135 ; RC[3] must hold the I/O base address of the MS780-E used
            ;3136 ;
            ;3137 ;
            ;3138 ; Outputs:
            ;3139 ; -------
            ;3140 ;
            ;3141 ; No specific outputs.
            ;3142 ;
            ;3143 ;
            ;3144 ; Side Effects:
            ;3145 ; ------------
            ;3146 ;
            ;3147 ; The contents of the following registers will be lost:
            ;3148 ;
            ;3149 ; VA, D, SC, Q
            ;3150 ;
            ;3151 ;--
            ;3152 ;*****************************************************************
            ;3153 =0
            ;3154 SETUP_MS780_RDS:
U 109E, 0000,003D,0180,0800,0000,1249  ;3155  CALL[MS780E.CLEANUP]  ; Make sure first that memory CSRs
            ;3156       ; ...are clear
U 109F, 0818,0038,B180,0800,0000,1264  ;3157  D_K[.4000]   ; Generate address 4020
U 1264, 0819,0030,7580,0800,0000,1265  ;3158  D_D.OR.K[.20]   ; ...
U 1265, 0001,003C,01E0,0800,0200,1266  ;3159  VA_D,Q_D   ; Get the address to the VA and
            ;3160       ; ...also save it in the Q reg
U 1266, 0818,0038,0D80,0800,0000,1267  ;3161  D_K[.3]    ; Get data to to force error (RDS)
U 1267, 0000,003C,0180,A800,0000,1268  ;3162  CACHE.P_D[LONG]   ; Write the data to location 4020
            ;3163  D_0,     ; Get data fro upper longword
U 1268, 0F00,003C,0180,0803,0000,1269  ;3164   VA_VA+4   ; Point VA to upper longword
U 1269, 0000,003C,0180,A800,0000,126A  ;3165  CACHE.P_D[LONG]   ; Initialize the upper longword
U 126A, 0000,003C,F580,0800,0084,726B  ;3166  SC_K[.A]   ; Prepare to set ECC bypass mode
            ;3167       ; ...in CNFG Reg B and ECC substitute
```

```
            ;3168       ; ...bits to x0C
U 126B, 0803,001C,0180,0800,0000,126C  ;3169  D_NOT.MASK    ; D = 400
U 126C, 0819,0030,8580,0800,0000,126D  ;3170  D_D.OR.K[.C]   ; D = 40C
U 126D, 0010,0038,0180,0918,0200,126E  ;3171  VA_RC[3]     ; Point VA to CNFG B
U 126E, 0000,003C,0180,0803,0000,126F  ;3172  VA_VA+4      ; ...
U 126F, 0000,003C,0180,A800,0000,1270  ;3173  CACHE.P_D[LONG]   ; Write the data to CNFG Reg B
U 1270, 0000,003E,0180,0800,0000,0001  ;3174  RETURN1      ; Return to caller
            ;3175
            ;3176
```

```
;3177  .PAGE
;3178  .TOC "   SET UP MS780-E CRD"
;3179 ;*************************************************************
;3180 ;+*
;3181 ;
;3182 ; Functional Description:
;3183 ; ----------------------
;3184 ;
;3185 ; This subtest sets up the memory so that a CRD will
;3186 ; take place when location 4020 is read.
;3187 ; The MS780-E is placed in Diagnostic mode 2 (ECC Bypass)
;3188 ; and locations 4020 and 4024 are intialized to x01 and
;3189 ; 0 respectively.
;3190 ;
;3191 ;
;3192 ; Calling Constraint: =0
;3193 ; ------------------
;3194 ;
;3195 ;
;3196 ; Inputs:
;3197 ; ------
;3198 ;
;3199 ; RC[3] must hold the I/O base address of the MS780-E used
;3200 ;
;3201 ;
;3202 ; Outputs:
;3203 ; -------
;3204 ;
;3205 ; No specific outputs.
;3206 ;
;3207 ;
;3208 ; Side Effects:
;3209 ; ------------
;3210 ;
;3211 ; The contents of the following registers will be lost:
;3212 ;
;3213 ; VA, D, SC, Q
;3214 ;
;3215 ;--
;3216 ;*************************************************************
;3217 =0
;3218 SETUP_MS780_CRD:
U 10A0, 0000,003D,0180,0800,0000,1249  ;3219  CALL[MS780E.CLEANUP]  ; Make sure first that memory CSRs
        ;3220          ; ...are clear
U 10A1, 0818,0038,B180,0800,0000,1271  ;3221  D_K[.4000]   ; Generate address 4020
U 1271, 0819,0030,7580,0800,0000,1272  ;3222  D_D.OR.K[.20]   ; ...
U 1272, 0001,003C,01E0,0800,0200,1273  ;3223  VA_D,Q_D   ; Get the address to the VA and
        ;3224          ; ...also save it in the Q reg
U 1273, 0818,0038,0580,0800,0000,1274  ;3225  D_K[.1]    ; Get data to to force error (CRD)
U 1274, 0000,003C,0180,A800,0000,1275  ;3226  CACHE.P_D[LONG]   ; Write the data to location 4020
        ;3227  D_0,     ; Get data fro upper longword
U 1275, 0F00,003C,0180,0803,0000,1276  ;3228   VA_VA+4   ; Point VA to upper longword
U 1276, 0000,003C,0180,A800,0000,1277  ;3229  CACHE.P_D[LONG]   ; Initialize the upper longword
U 1277, 0000,003C,F580,0800,0084,7278  ;3230  SC_K[.A]   ; Prepare to set ECC bypass mode
        ;3231          ; ...in CNFG Reg B and ECC substitute
```

```
            ;3232       ; ...bits to x0C
U 1278. 0803.001C.0180.0800.0000.1279  ;3233  D_NOT.MASK    ; D = 400
U 1279. 0819.0030.8580.0800.0000.127A  ;3234  D_D.OR.K[.C]    ; D = 40C
U 127A. 001D.0038.0180.0918.0200.127B  ;3235  VA_RC[3]    ; Point VA to CNFG B
U 127B. 000D.003C.0180.0803.0000.127C  ;3236  VA_VA+4      ; ...
U 127C. 000D.003C.0180.A800.0000.127D  ;3237  CACHE.P_D[LONG]    ; Write the data to CNFG Reg B
U 127D. 000D.003E.0180.0800.0000.0001  ;3238  RETURN1     ; Return to caller
            ;3239
            ;3240
            ;3241
            ;3242
```

```
;3243  .PAGE "TEST 213 NEXUS CRD/RDS TEST"
;3244 ;=============================================================================
;3245 ;++
;3246 ;
;3247 ;
;3248 ; NEXUS CRD/RDS TEST
;3249 ;
;3250 ;
;3251 ; TEST DISCRIPTION
;3252 ;
;3253 ; THIS TEST TESTS THE CRD/RDS FAULT DETECTION LOGIC IN SBI NEXUS
;3254 ; EXCEPT THE CPU.
;3255 ;
;3256 ; THIS IS DONE BY PROGRAMMING THE NEXUS TO PERFORM A READ FROM MEMORY
;3257 ; WHILE THE MEMORY IS SETUP TO FORCE THE ERROR CONDITION.
;3258 ; THE NEXUS IS THEN CHECKED FOR THE CORRECT RESPONSE.
;3259 ;
;3260 ; ERROR LOGOUT
;3261 ;
;3262 ; DATA: ITEM 1 (SEE INDIVIDUAL SUB-TEST)
;3263 ;  ITEM 2 (SEE INDIVIDUAL SUB-TEST)
;3264 ;  DEVICE FLAG
;3265 ;    0 = DW750
;3266 ;    1 = RH750
;3267 ;    2 = CI750
;3268 ;    3 = DR780
;3269 ;  TR LEVEL OF ADAPTER UNDER TEST
;3270 ;  TR LEVEL OF MEMORY BEING USED
;3271 ;
;3272 ;--
;3273 ;*****************************************************************************
;3274 ;
;3275 ; SCRATCH PAD REQUIREMENT
;3276 ; ----------------------
;3277 ;
;3278 ; RA,B DESCRIPTION
;3279 ; ---- -----------
;3280 ;
;3281 ;   5 OFFSET TO CI780 LOCAL STORE LOCATION ^X900
;3282 ;   6 BASE ADDRESS OF CI780 MICRO-STORE ^X400
;3283 ;   F U-TRAPS FLAGS
;3284 ;
;3285 ;
;3286 ; RC DESCRIPTION
;3287 ; -- -----------
;3288 ;  0 CONFIGURATION A ADDRESS OF CURRENT TR ^X200XX000
;3289 ;  3 BASE ADDRESS OF MEMORY BEING USED ^X200YY000
;3290 ;  7 BASE ADDRESS OF DW780 UNIBUS SPACE
;3291 ;  8 BASE ADDRESS OF RH780 DRIVE 0  ^X200XX400
;3292 ;  A TR LEVEL OF MEMROY BEING USED
;3293 ;  B TR LEVEL OF ADAPTER UNDER TEST
;3294 ;  C DEVICE FLAG (SEE ABOVE)
;3295 ;  D SEE INDIVIDUAL TESTS
;3296 ;  E SEE INDIVIDUAL TESTS
;3297 ;  F NUMBER OF ARGUMENTS TO BE PASSED TO THE CONSOLE
```

```
            ;3298 ;
            ;3299 ;*****************************************************************************
            ;3300 =0
U 10A2. 0000,003D,0180,0800,0000,1109  ;3301 M06000: NEWTST
U 10A3. 0000,003C,0180,0800,0000,10A4  ;3302  NOP
U 10A4. 0000,003D,0180,0800,0000,11DD  ;3303 =0 CALL[INIT]
U 10A5. 0000,003C,0180,0800,0000,10A6  ;3304  NOP
U 10A6. 0000,003D,0180,0800,0000,1246  ;3305 =0 CALL[CONFIG.MS780E]  ; Try to configure a MS780-E
            ;3306  D_RC[0E],    ; Get I/O Base Address of Memory
U 10A7. 0810,0038,8580,0970,0084,727E  ;3307   SC_K[.C]  ; Prepare to extract TR level from it
U 127E. 0001,003C,0180,0998,0000,127F  ;3308  RC[3]_D     ; Save I/O base address in RC[3]
U 127F. 0019,0034,6180,09D0,0000,1280  ;3309  RC[0A]_D.AND.K[.F]  ; SAVE IT, START TEST
            ;3310 ADP020:
U 1280. 0F00,003C,8580,0800,0084,7281  ;3311  D_0,SC_K[.C]
U 1281. 0801,001C,0180,0800,0000,1282  ;3312  D_D.ORNOT.MASK
U 1282. 0000,003C,0980,0800,0084,B283  ;3313  SC_SC-K[.2]
U 1283. 0801,001C,0180,0800,0000,1284  ;3314  D_D.ORNOT.MASK
U 1284. 0001,003C,0180,09A0,0000,1285  ;3315  RC[4]_D
U 1285. 0018,0038,0580,09D8,0000,1286  ;3316  RC[0B]_K[.1]   ;SET 1ST TR LEVEL TO BE CHECKED
U 1286. 0F00,003C,8980,0800,0084,7287  ;3317  D_0,SC_K[.D]
U 1287. 0801,001C,6580,0800,0000,1288  ;3318  D_D.ORNOT.MASK,K[.10]
U 1288. 0000,003C,6580,0800,0084,9289  ;3319  SC_SC+K[.10]
U 1289. 0801,001C,0180,0980,0000,10A8  ;3320  RC[0]_ALU,D_D.ORNOT.MASK
            ;3321  ;
            ;3322  ;=============================================================================
            ;3323  ;+
            ;3324  ;FIND THE ADAPTER CODE, IF REGISTER RESPONDS
            ;3325  ;
            ;3326  ; ERROR LOGOUT
            ;3327  ;
            ;3328  ; DATA: TR LEVEL BEING CHECKED
            ;3329  ;  IMAGE OF CONFIGURATION REGISTER READ
            ;3330  ;  DEVICE FLAG,0=UBA,1=MBA,2=YY780
            ;3331  ;  TR LEVEL OF ADAPTER UNDER TEST
            ;3332  ;  TR LEVEL OF MEMORY BEING USED
            ;3333  ;  UNDEFINED
            ;3334  ;-
            ;3335  ;=============================================================================
            ;3336  ;
            ;3337 =0
U 10A8. 0000,003D,0180,0800,0000,11DD  ;3338 CLN000: CALL[INIT]
U 10A9. 0003,003C,0180,09F8,0000,10AA  ;3339  RC[0F]_0    ; FORCE SUBTEST COUNTER TO ZERO
U 10AA. 0010,0039,D580,0900,0284,718D  ;3340 =0 VA_RC[0],SETRCF[.6]  ;SET ADDRESS,SET # ARG TO
U 10AB. 0F00,003C,8980,0800,0084,728A  ;3341  D_0,SC_K[.D]   ;CATCH A TIME OUT
U 128A. 0801,001C,0180,0AF8,0000,128B  ;3342  R[0F]_ALU,D_D.ORNOT.MASK
U 128B. 0000,003C,0180,C800,0000,128C  ;3343  MCT/READ.P    ;READ CONFIGURATION REGISTER
U 128C. 0001,003C,0180,09E8,0000,128D  ;3344  RC[0D]_D    ;SAVE THE DATA READ
U 128D. 0000,003C,0180,0A78,0010,128E  ;3345  ALU_R[0F],CLK.UBCC  ;DID IT TIMEOUT
U 128E. 0000,013C,0180,0800,0000,10AC  ;3346  Z?
U 10AC. 0000,003C,0180,0800,0000,1040  ;3347 =0 J/CLN010    ;NO TIMEOUT
U 10AD. 0000,003C,0180,0800,0000,132C  ;3348  J/CLN900    ;TIMED OUT , GO CHECK FOR END OF LOOOP
            ;3349 =00
U 1040. 0000,003D,0180,0800,0000,11C6  ;3350 CLN010: CALL[NOINTR]
U 1041. 0000,003D,0180,0800,0000,116A  ;3351  ERROR1     ;UNEXPECTED INTERRUPT !!!!
U 1042. 0810,0038,0180,0958,0000,1043  ;3352  D_RC[0B]    ;GET TR LEVEL UNDER TEST
```

```
U 1043. 0001,003C,0180,09F0,0000,128F   ;3353  RC[0E]_D    ;SET IT TO CONSOLE
U 128F. 0810,0038,0180,0968,0000,1290   ;3354  D_RC[0D]    ;GET CONFIG REG A DATA
U 1290. 0200,003C,0180,0800,0000,1291   ;3355  D_D.RIGHT2   ; SETUP TO TEST BITS<7:3>
U 1291. 0000,003C,01E0,0800,0000,1292   ;3356  Q_D
U 1292. 0200,003C,0180,0800,0000,1293   ;3357  D_D.RIGHT2   ; ...
U 1293. 0C00,193C,0180,0800,0000,1017   ;3358  D3?,D_Q     ; BIT 7 SET?
U 1017. 0000,003C,0180,0800,0000,1294   ;3359 =0111 J/CLN020    ; BRANCH IF NO
U 101F. 0000,003C,0180,0800,0000,132C   ;3360  J/CLN900    ; UNKNOWN DEVICE
U 1294. 0600,003C,0180,0800,0000,1295   ;3361 CLN020: D_D.RIGHT    ; GET BITS<6:3> INTO <3:0> POSITION
U 1295. 0000,193C,0180,0800,0000,1020   ;3362  D3-0?     ; BRANCH ON BITS<6:3>
U 1020. 0000,003C,0180,0800,0000,132C   ;3363 =0000 J/CLN900    ; UNKNOWN DEVICE
U 1021. 0000,003C,0180,0800,0000,132C   ;3364  J/CLN900    ; MS780
U 1022. 0000,003C,0180,0800,0000,132C   ;3365  J/CLN900    ; MS780
U 1023. 0000,003C,0180,0800,0000,132C   ;3366  J/CLN900    ; UNKNOWN
U 1024. 0018,0038,0580,09E0,0000,12BC   ;3367  RC[0C]_K[.1],J/CLN050   ; RH780
U 1025. 0018,0038,1980,09E0,0000,12B3   ;3368  RC[0C]_K[ZERO],J/CLN030   ; DW780
U 1026. 0018,0038,0D80,09E0,0000,12BE   ;3369  RC[0C]_K[.3],J/START_TEST ; DR780
U 1027. 0018,0038,0980,09E0,0000,10AE   ;3370  RC[0C]_K[.2],J/CLN040   ; CI780
U 1028. 0000,003C,0180,0800,0000,132C   ;3371  J/CLN900    ; MA780
U 1029. 0000,003C,0180,0800,0000,132C   ;3372  J/CLN900    ; UNKNOWN
U 102A. 0000,003C,0180,0800,0000,132C   ;3373  J/CLN900    ; UNKNOWN
U 102B. 0000,003C,0180,0800,0000,132C   ;3374  J/CLN900    ; UNKNOWN
U 102C. 0000,003C,0180,0800,0000,132C   ;3375  J/CLN900    ; MS780
U 102D. 0000,003C,0180,0800,0000,132C   ;3376  J/CLN900    ; MS780
U 102E. 0000,003C,0180,0800,0000,132C   ;3377  J/CLN900    ; MS780
U 102F. 0000,003C,0180,0800,0000,132C   ;3378  J/CLN900    ; MS780
          ;3379 ;
          ;3380 ; CI780. LOAD MICRO-CODE TO PERFORM READS FROM MEMORY AND INIT
          ;3381 ; THE DEVICE.
          ;3382 ;
          ;3383 ; FOLLOWING IS THE MICRO-CODE THAT GETS LOADED INTO THE CI780
          ;3384 ; TO PERFORM MEMORY READS:
          ;3385 ;* 50
          ;3386 ;$ 7401,00C1,DF20,0000 ;400: SYNC_SET,JMP[401]
          ;3387 ;$ F000,020B,5F20,0000 ;401: XBCAR_LS[CABUFA]
          ;3388 ;$ F000,00C1,1F20,0000 ;402: NULLUW
          ;3389 ;$ F010,00C1,5F20,0000 ;403: XBUS_A_GO
          ;3390 ;$ 0408,00C0,5F20,0000 ;404: BR_MEM_A[408]
          ;3391 ;$ F000,00C1,1F20,0000 ;405: NULLUW
          ;3392 ;$ F000,00C1,1F20,0000 ;406: NULLUW
          ;3393 ;$ 0408,00C0,5F20,0000 ;407: BR_MEM_A[408]
          ;3394 ;$ 0408,00C0,5F20,0000 ;408: BR_MEM_A[408]
          ;3395 ;$ 0408,00C0,5F20,0000 ;409: BR_MEM_A[408]
          ;3396 ;$ 7410,00C1,5F20,0000 ;40A: JMP[410]
          ;3397 ;$ 7410,00C1,5F20,0000 ;40B: JMP[410]
          ;3398 ;$ 0408,00C0,5F20,0000 ;40C: BR_MEM_A[408]
          ;3399 ;$ 0408,00C0,5F20,0000 ;40D: BR_MEM_A[408]
          ;3400 ;$ 7411,00C1,1F20,0000 ;40E: JMP[411]
          ;3401 ;$ 7411,00C1,1F20,0000 ;40F: JMP[411]
          ;3402 ;$ 7411,40B9,5F20,0000 ;410: LS[40]_[40],JMP[411]
          ;3403 ;$ F000,01B9,5F20,0000 ;411: LS[01]_[01]
          ;3404 ;$ 7412,00C1,1F20,0000 ;412: JMP[412]
          ;3405 ;$ F000,00C1,1F20,0000 ;413: NULLUW
          ;3406 ;$ F000,00C1,1F20,0000 ;414: NULLUW
          ;3407 ;
```

```
            ;3408 ;
            ;3409 =0
U 10AE. 0000.003D.0180.0800.0000.1195  ;3410 CLN040: CALL[INIT_CI780]  ; INITIALIZE THE CI780
            ;3411    ;
            ;3412    ; LOAD THE MICRO CODE INTO THE CI750
            ;3413    ;
U 10AF. 0F00.003C.0180.0800.0000.1296  ;3414  D_0    ; FIRST ENABLE CACHE
U 1296. 0000.003C.7580.3C00.0000.1297  ;3415  ID[MAINT]_D   ; ...
U 1297. 0810.0038.0180.0900.0000.1298  ;3416  D_RC[0]    ; GET BASE ADDRESS OF CI780
U 1298. 0019.0030.2180.0A88.0000.10B0  ;3417  R[1]_ALU,ALU_D.OR.K[.14] ; GENERATE ADDRESS OF 'MADR' REG
U 10B0. 0818.0039.3180.0800.0000.1172  ;3418 =0 D_K[.40],CALL[DC0]  ; GENERATE ADDRESS OF MICRO STORE
U 10B1. 0001.003C.8580.0A90.0084.7299  ;3419  R[2]_D,SC_K[.C]   ; SAVE
U 1299. 0001.003C.0180.0AB0.0000.129A  ;3420  R[6]_D    ; ...
U 129A. 0001.001C.0180.0A98.0000.129B  ;3421  R[3]_ALU,ALU_D.ORNOT.MASK ; UPPER LONG WORD OF MICRO STORE
U 129B. 0818.0038.3580.0800.0000.129C  ;3422  D_K[.50]   ; GET ADDRESS OF DATA TO LOAD
U 129C. 0001.003C.0580.0AA0.0104.729D  ;3423  R[4]_D,FE_K[.1]   ; GENERATE ADDRESS OF DATA TO LOAD
U 129D. 0000.003C.2180.0800.0084.729E  ;3424  SC_K[.14]   ; SET THE LOOP COUNT
            ;3425 CI780_LOAD_LOOP:
U 129E. 0800.003C.0180.0A20.0200.129F  ;3426  VA_R[4],D_ALU   ; GET ADDRESS OF DATA TO LOAD
U 129F. 0019.0014.1180.0AA0.0000.12A0  ;3427  R[4]_D+K[.4]   ; INCREMENT AND SAVE
U 12A0. 0000.003C.0180.C800.0000.12A1  ;3428  D[LONG]_CACHE.P   ; GET LONG WORD TO LOAD
U 12A1. 0800.003C.01E0.0A10.0000.12A2  ;3429  Q_D,D_R[2]   ; SAVE IN Q AND GET CTRL STORE ADDRESS
U 12A2. 0000.003C.0180.0A08.0200.12A3  ;3430  VA_R[1]    ; SET ADDRESS OF 'MADR' REG
U 12A3. 0000.003C.0180.A800.0000.12A4  ;3431  CACHE.P_D[LONG]   ; WRITE CONTROL STORE ADDRESS
U 12A4. 0C00.003C.0180.0803.0000.12A5  ;3432  VA_VA+4,D_Q   ; SET ADDRESS OF 'MDATR' AND GET DATA
U 12A5. 0000.003C.0180.A800.0000.12A6  ;3433  CACHE.P_D[LONG]   ; WRITE THE DATA
U 12A6. 0000.003C.0180.0A20.0200.12A7  ;3434  VA_R[4]    ; GET ADDRESS OF NEXT DATA TO LOAD
U 12A7. 0000.003C.0180.C800.0000.12A8  ;3435  D[LONG]_CACHE.P   ; GET DATA TO LOAD
U 12A8. 0800.003C.01E0.0A18.0000.12A9  ;3436  Q_D,D_R[3]   ; GET CTRL STORE ADDRESS
U 12A9. 0000.003C.0180.0A08.0200.12AB  ;3437  VA_R[1]    ; SET ADDRESS OF 'MADR' REG
U 12AB. 0000.003C.0180.A800.0000.12AC  ;3438  CACHE.P_D[LONG]   ; WRITE CTRL STORE ADDRESS (1400)
U 12AC. 0C00.003C.0180.0803.0000.12AD  ;3439  VA_VA+4,D_Q   ; GET DATA TO LOAD
U 12AD. 0000.003C.0180.A800.0000.12AE  ;3440  CACHE.P_D[LONG]   ; WRITE UPPER HALF OF DATA
U 12AE. 0800.0C3C.0180.0A20.0000.1003  ;3441  SC.NE.0?,D_R[4]   ; LOADED ALL MICRO WORDS YET?
U 1003. 0000.003C.0180.0800.0000.10B2  ;3442 =011 J/CI780_LOAD_DONE  ; BRANCH IF YES
            ;3443  SC_SC-FE.    ; DECREMENT LOOP COUNTER AND
U 1007. 0019.0014.1180.0AA0.0080.B2AF  ;3444       R[4]_D+K[.4]  ; INCREMENT LOAD DATA ADDRESS
U 12AF. 0800.003C.0180.0A18.0000.12B0  ;3445  D_R[3]
U 12B0. 0019.0014.0580.0A98.0000.12B1  ;3446  R[3]_D+K[.1]   ; INCREMENT CONTROL STORE ADDRESS
U 12B1. 0800.003C.0180.0A10.0000.12B2  ;3447  D_R[2]
U 12B2. 0019.0014.0580.0A90.0000.129E  ;3448  R[2]_D+K[.1],J/CI780_LOAD_LOOP ; ...
            ;3449 =0
            ;3450 CI780_LOAD_DONE:
U 10B2. 0000.003D.0180.0800.0000.116E  ;3451  CALL[DISABLE.CACHE]  ; DISABLE CACHE
U 10B3. 0000.003C.0180.0800.0000.12BE  ;3452  J/START_TEST   ; START THE TEST
            ;3453 ;
            ;3454 ; UBA. CALCULATE STARTING ADDRESS OF UNIBUS SPACE
            ;3455 ;
U 12B3. 0810.0038.0180.0968.0000.12B4  ;3456 CLN030: D_RC[0D]   ;GET CONFIG REG A IMAGE THAT WAS READ
U 12B4. 0819.0034.0D80.0800.0000.12B5  ;3457  D_D[AND]K[.3]   ;FIND UBA BASE ADDRESS FOR SBI I/O SPACE
U 12B5. 0819.0030.11F8.0800.0000.12B6  ;3458  D_D[OR]K[.4],Q_0
U 12B6. 0000.003C.2180.0800.0084.72B7  ;3459  SC_K[.14]
U 12B7. 0000.003C.0980.0800.0084.B2B8  ;3460  SC_SC-K[.2]
U 12B8. 0D00.003C.D980.0800.0000.12B9  ;3461  D_DAL.SC.K[.9]
U 12B9. 0000.003C.D980.0800.0084.92BA  ;3462  SC_SC+K[.9]
```

```
U 12BA, 0000,003C,0980,0800,0084,92BB  ;3463  SC_SC+K[.2]
U 12BB, 0801,001C,0180,09B8,0000,12BE  ;3464  RC[7]_ALU,D_D.ORNOT.MASK,J/START_TEST
          ;3465
          ;3466 ;
          ;3467 ; RH780. GENERATE ADDRESS OF DRIVE REGISTER
          ;3468 ;
U 12BC, 0810,0038,F580,0900,0084,72BD  ;3469 CLN050: D_RC[0],SC_K[.A]   ; GET CONFIG REGISTER ADDRESS
          ;3470  R[8]_ALU,ALU_D.ORNOT.MASK, ; GENERATE ADDRESS OF DRIVE 0
U 12BD, 0001,001C,0180,0AC0,0000,12BE  ;3471       J/START_TEST
          ;3472
          ;3473 START_TEST:
U 12BE, 001B,0010,1180,09F8,0000,10B4  ;3474  RC[0F]_ALU,ALU_0+K[.4]+1 ; SET NUMBER OF DATA ARG'S TO 2
U 10B4, 0000,003D,0180,0800,0000,1187  ;3475 =0 SUBTEST     ; FORCE SUBTEST COUNTER TO 1
U 10B5, 0000,003C,0180,0800,0000,10B6  ;3476  NOP
          ;3477 ;
          ;3478 ;*************************************************************************
          ;3479 ;*
          ;3480 ;** SUBTEST 2 **
          ;3481 ;
          ;3482 ; CHECK THAT THE CI780 CAN DETECT RDS ERRORS FROM THE MEMORY
          ;3483 ; MICRO CODE HAS BEEN LOADED INTO THE CI780 TO PERFORM READS FROM MEMORY.
          ;3484 ; THIS SUBTEST WILL CAUSE THIS MICRO CODE TO READ MEMORY LOCATION 4020(X).
          ;3485 ; THE MEMORY WILL BE SETUP TO CAUSE AN RDS TAG TO BE RETURNED TO THE CI780.
          ;3486 ; THEN, THE CI780 IS CHECKED TO SEE THAT THE HARDWARE SET BIT 17 IN THE
          ;3487 ; CONFIGURATION REGISTER AND THAT THE CI780 MICRO-CODE DETECTED THE ERROR.
          ;3488 ;
          ;3489 ; DATA: EXPECTED CI780 CONFIG. REG OR LOCAL STORE
          ;3490 ; RECEIVED CI780 CONFIG. REG OR LOCAL STORE
          ;3491 ;  SEE ABOVE
          ;3492 ;      .
          ;3493 ;      .
          ;3494 ;      .
          ;3495 ;
U 10B6, 0000,003D,0180,0800,0000,1187  ;3496 =0 SUBTEST
U 10B7, 0810,0038,0180,0960,0000,12BF  ;3497  D_RC[0C]    ; GET ADAPTER TYPE CODE
U 12BF, 0019,0000,0980,0800,0010,12C0  ;3498  ALU_D-K[.2],CLK.UBCC  ; IS IT A CI780?
U 12C0, 0000,013C,0180,0800,0000,10B8  ;3499  Z?
U 10B8, 0000,003C,0180,0800,0000,10CF  ;3500 =0 J/NEXUS_TEST_3    ; BRANCH IF NO
U 10B9, 0000,003C,0180,0800,0000,10BA  ;3501  NOP
          ;3502      ;
          ;3503      ; THIS IS A CI780. INIT THE CI780. THEN, SETUP THE MEMORY TO
          ;3504      ; FORCE THE ERROR.
          ;3505      ;
U 10BA, 0000,003D,0180,0800,0000,1169  ;3506 =0 ERLOOP
U 10BB, 0000,003C,0180,0800,0000,10BC  ;3507  NOP
U 10BC, 0000,003D,0180,0800,0000,1195  ;3508 =0 CALL[INIT_CI780]  ; INIT THE CI780
U 10BD, 0000,003C,0180,0800,0000,10BE  ;3509  NOP
U 10BE, 0818,0039,4180,0800,0000,117A  ;3510 =0 D_K[.80],CALL[DC8]  ; GENERATE OFFSET FOR 'LS_CA' REG
U 10BF, 0811,0030,0180,0900,0000,12C1  ;3511  D_D.OR.RC[0]    ; ...
U 12C1, 0001,003C,ED80,0800,0284,72C2  ;3512  VA_ALU,ALU_D,SC_K[.1B]  ; SET ADDRESS OF 'LS_CA' REG
U 12C2, 0000,003C,0180,0800,0080,D2C3  ;3513  SC_SC+1     ; SC = 1C(X)
U 12C3, 0803,001C,8580,0800,0084,72C4  ;3514  D_NOT.MASK,SC_K[.C]  ; GET SBI ADDRESS AND FUNCTION
U 12C4, 0801,001C,0180,0800,0000,12C5  ;3515  D_D.ORNOT.MASK    ; (BYTE ADDRESS IS 4020)
U 12C5, 0819,0030,0180,0800,0000,12C6  ;3516  D_D.OR.K[.8]    ; EQUALS 10001008
U 12C6, 0000,003C,0180,A800,0000,10C0  ;3517  CACHE.P_D[LONG]    ; PASS TO CI780 MICRO-CODE
```

```
U 10C0. 0000.003D.0180.0800.0000.109E  ;3518 =0 CALL[SETUP_MS780_RDS]  ; SETUP MEMORY TO FORCE THE ERROR
          ;3519     ;
          ;3520     ; EVERYTHING IS SETUP. START THE CI780 READ FROM MEMORY
          ;3521     ;
U 10C1. 0810.0038.0180.0900.0000.12C7  ;3522  D_RC[0]
U 12C7. 0019.0030.2180.0800.0200.12C8  ;3523  VA_ALU,ALU_D.OR.K[.14]  ; GET 'MADR' ADDRESS
U 12C8. 0800.003C.0180.0A30.0084.72C9  ;3524  D_R[6],SC_K[.8]   ; GET ADDRESS OF START OF MICRO CODE
U 12C9. 0000.003C.0180.A800.0000.12CA  ;3525  CACHE.P_D[LONG]   ; SEND IT TO CI780
U 12CA. 0818.0038.D9F8.0800.0000.12CB  ;3526  D_K[.9],Q_0   ; GENERATE OFFSET OF 'PICR' REG
U 12CB. 0D00.003C.0180.0800.0000.12CC  ;3527  D_DAL.SC   ; ...
U 12CC. 0001.003C.0180.0AA8.0000.12CD  ;3528  R[5]_D    ; SAVE PARTIAL OFFSET (900)
U 12CD. 0F11.0030.01E0.0900.0200.12CE  ;3529  VA_ALU,ALU_D[GR]RC[0],Q_D,D_0 ; CLEAR LOCAL STORE LOCATION THAT
U 12CE. 0000.003C.0180.A800.0000.12CF  ;3530  CACHE.P_D[LONG]   ; GETS THE RESULT.
U 12CF. 0819.2030.E980.0800.0000.12D0  ;3531  D_Q.OR.K[.24]   ; ...
U 12D0. 0011.0030.0180.0900.0200.12D1  ;3532  VA_ALU,ALU_D[OR]RC[0]  ; SET ADDRESS OF 'PICR' REG
U 12D1. 0818.0038.0580.0800.0000.12D2  ;3533  D_K[.1]
U 12D2. 0000.003C.0180.A800.0000.10C2  ;3534  CACHE.P_D[LONG]   ; START THE CI780 READ
          ;3535 =0 D_K[.FF],   ; WAIT FOR APPROXIMATELY 100 MICRO-SEC
U 10C2. 0818.0039.4980.0800.0000.1205  ;3536   CALL[WAIT.LOOP]
U 10C3. 0000.003C.0180.0800.0000.10C4  ;3537  NOP
U 10C4. 0000.003D.0180.0800.0000.1249  ;3538 =0 CALL[MS780E.CLEANUP]  ; CLEANUP MEMORY
U 10C5. 0810.0038.D580.0900.0084.72D3  ;3539  D_RC[0],SC_K[.6]
U 12D3. 000D.0030.0180.0A28.0200.12D4  ;3540  VA_ALU,ALU_D[OR]R[5]  ; LOAD ADDRESS OF LOCAL STORE
U 12D4. 0000.003C.0180.C800.0000.12D5  ;3541  D[LONG]_CACHE.P   ; READ CI780 LOCAL STORE
U 12D5. 0000.003C.0180.0800.0000.12D6  ;3542  NOP
U 12D6. 0001.003C.0180.09E8.0010.12D7  ;3543  RC[0D]_ALU,ALU_D,CLK.UBCC ; IS RDS BIT SET?
U 12D7. 0003.011C.0180.09F0.0000.10C6  ;3544  Z?,RC[0E]_NOT.MASK  ; IS IT SET?
U 10C6. 0000.003C.0180.0800.0000.10C9  ;3545 =0 J/CI780_RDS   ; YES
U 10C7. 0000.003C.0180.0800.0000.10C8  ;3546  NOP
U 10C8. 0000.003D.0180.0800.0000.116C  ;3547 =0 ERROR2
          ;3548 ;
          ;3549 ; CHECK THAT BIT 17 IN CONFIG REG A SET
          ;3550 ;
          ;3551 CI780_RDS:
U 10C9. 0010.0038.6580.0900.0284.72D8  ;3552  VA_ALU,ALU_RC[0],SC_K[.10] ; LOAD REGISTER CI780 BASE ADDRESS
U 12D8. 0000.003C.0180.C800.0080.D2D9  ;3553  D[LONG]_CACHE.P,SC_SC+1  ; READ CONFIG REG
          ;3554  RC[0D]_ALU,ALU_D.AND.MASK, ; SEE IF RDS BIT IS SET
U 12D9. 0001.0034.0180.09E8.0010.12DA  ;3555       CLK.UBCC   ; IS IT SET?
U 12DA. 0003.011C.0180.09F0.0000.10CA  ;3556  Z?,RC[0E]_NOT.MASK
U 10CA. 0000.003C.0180.0800.0000.10CD  ;3557 =0 J/CI780_RDS_CLR   ; BRANCH IF YES
U 10CB. 0000.003C.0180.0800.0000.10CC  ;3558  NOP
U 10CC. 0000.003D.0180.0800.0000.116C  ;3559 =0 ERROR2    ; NO
          ;3560 ;
          ;3561 ; CHECK THAT BIT 17 IN CONFIG REG A CLEARS
          ;3562 ;
          ;3563 CI780_RDS_CLR:
U 10CD. 0010.0038.6580.0900.0284.72DB  ;3564  VA_ALU,ALU_RC[0],SC_K[.10] ; RELOAD CONFIG REG A ADDRESS
U 12DB. 0000.003C.0180.0800.0080.D2DC  ;3565  SC_SC+1
U 12DC. 0803.001C.01C0.0800.0000.12DD  ;3566  D_NOT.MASK,Q_ALU  ; GET DATA TO TRY AND CLEAR THE BIT
U 12DD. 0000.003C.0180.A800.0000.12DE  ;3567  CACHE.P_D[LONG]   ; TRY AND CLEAR THE BIT
U 12DE. 0000.003C.0180.C800.0000.12DF  ;3568  D[LONG]_CACHE.P   ; READ THE BIT
U 12DF. 001D.0034.0180.09E8.0010.12E0  ;3569  ALU_D.AND.Q,RC[0D]_ALU,CLK.UBCC ; DID BIT CLEAR?
U 12E0. 0003.013C.0180.09F0.0000.10CE  ;3570  Z?,RC[0E]_0
U 10CE. 0000.003D.0180.0800.0000.116C  ;3571 =0 ERROR2    ; RDS BIT DID NOT CLEAR IN CONFIG REG
          ;3572 NEXUS_TEST_3:
```

```
U 10CF. 0000.003C.0180.0800.0000.10D0  ;3573  NOP
          ;3574
          ;3575 ;////////////////////////////////////////////////////////////////////////////
          ;3576 ;
          ;3577 ; SUBTEST 3
          ;3578 ;
          ;3579 ; THIS SUBTEST CHECKS THE RDS BIT IN THE UNIBUS ADAPTER. THIS IS DONE
          ;3580 ; BE PERFORMING A READ TO THE UNIBUS THAT GETS MAPPED INTO A READ TO
          ;3581 ; MAIN MEMORY. THE MEMORY WILL HAVE THE RDS ERROR FORCED.
          ;3582 ;
          ;3583 ; THE READ IS PERFORMED TO UNIBUS ADDRESS 20(X) WHICH MAPS THRU MAP REGISTER 0.
          ;3584 ; MAP REGISTER 0 WILL CONTAIN 80000010(X) TO GENERATE AN SBI ADDRESS OF
          ;3585 ; 4020(X).
          ;3586 ;
          ;3587 ; DATA: EXPECTED DW780 STATUS REGISTER (BIT 9 ONLY)
          ;3588 ; RECEIVED DW780 STATUS REGISTER (BIT 9 ONLY)
          ;3589 ;   SEE ABOVE
          ;3590 ;       .
          ;3591 ;       .
          ;3592 ;
U 10D0. 0000.003D.0180.0800.0000.1187  ;3593 =0 SUBTEST
U 10D1. 0010.0038.0180.0960.0010.12E1  ;3594  ALU_RC[0C],CLK.UBCC  ; IS THIS DEVICE A DW780?
U 12E1. 0000.013C.0180.0800.0000.10D2  ;3595  Z?
U 10D2. 0000.003C.0180.0800.0000.10E7  ;3596 =0 J/NEXUS_TEST_4   ; BRANCH IF NO
U 10D3. 0000.003C.0180.0800.0000.10D4  ;3597  NOP
U 10D4. 0000.003D.0180.0800.0000.1169  ;3598 =0 ERLOOP
U 10D5. 0000.003C.0180.0800.0000.10D6  ;3599  NOP
U 10D6. 0000.003D.0180.0800.0000.11DD  ;3600 =0 CALL[INIT]   ; CLEANUP THE CPU AND ADAPTER
U 10D7. 0000.003C.0180.0800.0000.10D8  ;3601  NOP
U 10D8. 0000.003D.0180.0800.0000.1191  ;3602 =0 CALL[INIT_DW780]  ; INIT THE UBA
U 10D9. 0000.003C.0180.0800.0000.10DA  ;3603  NOP
U 10DA. 0000.003D.0180.0800.0000.11F9  ;3604 =0 CALL[WAITFORUNIBUS]  ; WAIT FOR UNIBUS TO COME UP
U 10DB. 0000.003C.0180.0800.0000.10DC  ;3605  NOP
U 10DC. 0000.003D.0180.0800.0000.109E  ;3606 =0 CALL[SETUP_MS780_RDS]  ; SETUP THE MEMORY TO FORCE THE ERROR
U 10DD. 0000.003C.0180.0800.0000.10DE  ;3607  NOP
U 10DE. 0000.003D.0180.0800.0000.119F  ;3608 =0 CALL[DW780_READ]  ; CAUSE THE DW780 TO DO A READ
U 10DF. 0000.003C.0180.0800.0000.10E0  ;3609  NOP
U 10E0. 0000.003D.0180.0800.0000.1249  ;3610 =0 CALL[MS780E.CLEANUP]  ; CLEANUP THE MEMORY
U 10E1. 0810.0038.0180.0900.0000.12E2  ;3611  D_RC[0]
U 12E2. 0019.0030.0180.0800.0200.12E3  ;3612  VA_ALU,ALU_D.OR.K[.8]  ; LOAD ADDRESS OF UNIBUS STATUS REG
U 12E3. 0000.003C.D980.C800.0084.72E4  ;3613  D[LONG]_CACHE.P,SC_K[.9] ; GET THE STATUS REGISTER
          ;3614  ALU_D.ANDNOT.MASK,RC[0D]_ALU,
U 12E4. 0001.0024.0180.09E8.0010.12E5  ;3615      CLK.UBCC   ; IS RDS BIT SET?
U 12E5. 0003.011C.0180.09F0.0000.10E2  ;3616  Z?,RC[0E]_NOT.MASK
U 10E2. 0000.003C.0180.0800.0000.10E5  ;3617 =0 J/DW780_RDS_CLR   ; BRANCH IF YES
U 10E3. 0000.003C.0180.0800.0000.10E4  ;3618  NOP
U 10E4. 0000.003D.0180.0800.0000.116C  ;3619 =0 ERROR2    ; RDS BIT NOT SET
          ;3620 DW780_RDS_CLR:
U 10E5. 0810.0038.D980.0900.0084.72E6  ;3621  D_RC[0],SC_K[.9]
U 12E6. 0019.0030.0180.0800.0200.12E7  ;3622  VA_ALU,ALU_D.OR.K[.8]  ; LOAD ADDRESS OF UBA STATUS REG
U 12E7. 0803.001C.0180.0800.0000.12E8  ;3623  D_NOT.MASK
U 12E8. 0000.003C.0180.A800.0000.12E9  ;3624  CACHE.P_D[LONG]   ; TRY AND CLEAR RDS BIT
U 12E9. 0000.003C.0180.C800.0000.12EA  ;3625  D[LONG]_CACHE.P   ; READ THE STATUS REGISTER
          ;3626  ALU_D.ANDNOT.MASK,RC[0D]_ALU,
U 12EA. 0001.0024.0180.09E8.0010.12EB  ;3627      CLK.UBCC   ; DID RDS BIT CLEAR?
```

```
U 12EB. 0003.013C.0180.09F0.0000.10E6  ;3628  Z?,RC[0E]_0
U 10E6. 0000.003D.0180.0800.0000.116C  ;3629 =0 ERROR2    ; NO
        ;3630
        ;3631 NEXUS_TEST_4:
U 10E7. 0000.003C.0180.0800.0000.10E8  ;3632  NOP
        ;3633 ;////////////////////////////////////////////////////////////////////////
        ;3634 ;
        ;3635 ; SUBTEST 4
        ;3636 ;
        ;3637 ; RH780 RDS SUBTEST.
        ;3638 ;
        ;3639 ; THIS TEST SETS UP THE MEMORY TO FORCE AN RDS ERROR AND THEN PROGRAMS THE
        ;3640 ; RH780 TO PERFORM A READ TO THE MEMORY. THE STATUS BIT IN THE RH780 IS
        ;3641 ; THEN TESTED TO ENSURE IT DETECTED THE RDS.
        ;3642 ;
        ;3643 ; DATA: EXPECTED RH780 STATUS REGISTER (BIT 2 ONLY)
        ;3644 ; RECEIVED RH780 STATUS REGISTER (BIT 2 ONLY)
        ;3645 ;
U 10E8. 0000.003D.0180.0800.0000.1187  ;3646 =0 SUBTEST
U 10E9. 0810.0038.0180.0960.0000.12EC  ;3647  D_RC[0C]    ; GET DEVICE CODE
U 12EC. 0019.0000.0580.0800.0010.12ED  ;3648  ALU_D-K[.1],CLK.UBCC  ; RH780?
U 12ED. 0000.013C.0180.0800.0000.10EA  ;3649  Z?
U 10EA. 0000.003C.0180.0800.0000.10FB  ;3650 =0 J/NEXUS_TEST_5    ; NO
U 10EB. 0000.003C.0180.0800.0000.10EC  ;3651  NOP
U 10EC. 0000.003D.0180.0800.0000.1169  ;3652 =0 ERLOOP
U 10ED. 0000.003C.0180.0800.0000.10EE  ;3653  NOP
U 10EE. 0000.003D.0180.0800.0000.11DD  ;3654 =0 CALL[INIT]    ; INIT THE RH780
U 10EF. 0000.003C.0180.0800.0000.10F0  ;3655  NOP
U 10F0. 0000.003D.0180.0800.0000.109E  ;3656 =0 CALL[SETUP_MS780_RDS]  ; SETUP THE MEMORY
U 10F1. 0000.003C.0180.0800.0000.10F2  ;3657  NOP
U 10F2. 0000.003D.0180.0800.0000.104C  ;3658 =0 CALL[RH780_READ]  ; PERFORM AN RH780 READ
U 10F3. 0000.003C.0180.0800.0000.10F4  ;3659  NOP
U 10F4. 0000.003D.0180.0800.0000.1249  ;3660 =0 CALL[MS780E.CLEANUP]  ; CLEANUP THE MEMORY
U 10F5. 0010.0038.01C0.0900.0000.12EE  ;3661  Q_RC[0]     ; GET ADDRESS OF CONFIG REG
U 12EE. 0019.2030.0180.0800.0200.12EF  ;3662  VA_ALU,ALU_Q.OR.K[.8]  ; SET ADDRESS OF STATUS REGISTER
U 12EF. 0000.003C.0180.C800.0000.12F0  ;3663  D[LONG]_CACHE.P    ; READ THE STATUS REGISTER
U 12F0. 0819.0034.1180.0800.0010.12F1  ;3664  D_D.AND.K[.4],CLK.UBCC  ; MASK
U 12F1. 0018.0138.1180.09F0.0000.10F6  ;3665  Z?,RC[0E]_K[.4]    ; IS IT SET?
U 10F6. 0000.003C.0180.0800.0000.10F9  ;3666 =0 J/RH780_RDS_CLR    ; BRANCH IF YES
U 10F7. 0000.003C.0180.0800.0000.10F8  ;3667  NOP
U 10F8. 0001.003D.0180.09E8.0000.116C  ;3668 =0 ERROR2,RC[0D]_D    ; RDS BIT DID NOT SET
        ;3669 RH780_RDS_CLR:
U 10F9. 0010.0038.01C0.0900.0000.12F2  ;3670  Q_RC[0]     ; GET RH780 BASE ADDRESS
U 12F2. 0019.2030.0180.0800.0200.12F3  ;3671  VA_ALU,ALU_Q.OR.K[.8]  ; SET ADDRESS OF STATUS REG
U 12F3. 0818.0038.1180.0800.0000.12F4  ;3672  D_K[.4]     ; GET DATA TO CLEAR THE RDS BIT
U 12F4. 0000.003C.0180.A800.0000.12F5  ;3673  CACHE.P_D[LONG]    ; TRY AND CLEAR THE RDS BIT
U 12F5. 0000.003C.0180.C800.0000.12F6  ;3674  D[LONG]_CACHE.P    ; READ THE STATUS REG AGAIN
U 12F6. 0819.0034.1180.0800.0010.12F7  ;3675  D_D.AND.K[.4],CLK.UBCC  ; DID RDS BIT CLEAR?
U 12F7. 0003.013C.0180.09F0.0000.10FA  ;3676  Z?,RC[0E]_0    ; ...
U 10FA. 0001.003D.0180.09E8.0000.116C  ;3677 =0 ERROR2,RC[0D]_D    ; RDS BIT DID NOT CLEAR
        ;3678 NEXUS_TEST_5:
U 10FB. 0000.003C.0180.0800.0000.10FC  ;3679  NOP
        ;3680 ;////////////////////////////////////////////////////////////////////////
        ;3681 ;
        ;3682 ; SUBTEST 5
```

```
            ;3683 ;
            ;3684 ; CI780 CRD.
            ;3685 ;
            ;3686 ; THIS SUBTEST CAUSES THE CI780 TO PERFORM A READ
            ;3687 ; TO A MEMORY LOCATION THAT IS SETUP TO FORCE A CRD ERROR. BIT 16 IN THE
            ;3688 ; CONFIGURATION REGISTER IS THEN CHECKED TO DETERMINE IF THE CI780 RECEIVED
            ;3689 ; THE CRD TAG FROM MEMORY.
            ;3690 ;
            ;3691 ; DATA: EXPECTED CONFIGURATION REGISTER (BIT 16 ONLY)
            ;3692 ; RECEIVED CONFIGURATION REGISTER
            ;3693 ;   SEE ABOVE
            ;3694 ;       .
            ;3695 ;       .
            ;3696 ;       .
            ;3697 ;
U 10FC. 0000.003D.0180.0800.0000.1187  ;3698 =0 SUBTEST
U 10FD. 0810.0038.0180.0960.0000.12F8  ;3699  D_RC[0C]    ; CHECK IF THIS IS A CI780
U 12F8. 0019.0000.0980.0800.0010.12F9  ;3700  ALU_D-K[.2],CLK.UBCC  ; CI780?
U 12F9. 0000.013C.0180.0800.0000.10FE  ;3701  Z?
U 10FE. 0000.003C.0180.0800.0000.111F  ;3702 =0 J/NEXUS_TEST_6    ; NO
U 10FF. 0000.003C.0180.0800.0000.110A  ;3703  NOP
            ;3704      ;
            ;3705      ; THIS IS A CI780. INIT THE CI780. THEN, SETUP THE MEMORY TO
            ;3706      ; FORCE THE ERROR.
            ;3707      ;
U 110A. 0000.003D.0180.0800.0000.1169  ;3708 =0 ERLOOP
U 110B. 0000.003C.0180.0800.0000.1110  ;3709  NOP
U 1110. 0000.003D.0180.0800.0000.1195  ;3710 =0 CALL[INIT_CI780]  ; INIT THE CI780
U 1111. 0000.003C.0180.0800.0000.1112  ;3711  NOP
U 1112. 0818.0039.4180.0800.0000.117A  ;3712 =0 D_K[.80],CALL[DC8]  ; GENERATE OFFSET FOR 'LS_CA' REG
U 1113. 0811.0030.0180.0900.0000.12FA  ;3713  D_D.OR.RC[0]    ; ...
U 12FA. 0001.003C.ED80.0800.0284.72FB  ;3714  VA_ALU,ALU_D,SC_K[.1B]  ; SET ADDRESS OF 'LS_CA' REG
U 12FB. 0000.003C.0180.0800.0080.D2FC  ;3715  SC_SC+1     ; SC = 1C(X)
U 12FC. 0803.001C.8580.0800.0084.72FD  ;3716  D_NOT.MASK,SC_K[.C]  ; GET SBI ADDRESS AND FUNCTION
U 12FD. 0801.001C.0180.0800.0000.12FE  ;3717  D_D.ORNOT.MASK    ; (BYTE ADDRESS IS 4020)
U 12FE. 0819.0030.0180.0800.0000.12FF  ;3718  D_D.OR.K[.8]    ; EQUALS 10001008
U 12FF. 0000.003C.0180.A800.0000.1114  ;3719  CACHE.P_D[LONG]    ; PASS TO CI780 MICRO-CODE
U 1114. 0000.003D.0180.0800.0000.10A0  ;3720 =0 CALL[SETUP_MS780_CRD]  ; SETUP MEMORY TO FORCE THE ERROR
            ;3721      ;
            ;3722      ; EVERYTHING IS SETUP. START THE CI780 READ FROM MEMORY
            ;3723      ;
U 1115. 0810.0038.0180.0900.0000.1300  ;3724  D_RC[0]
U 1300. 0019.0030.2180.0800.0200.1301  ;3725  VA_ALU,ALU_D.OR.K[.14]  ; GET 'MADR' ADDRESS
U 1301. 0800.003C.0180.0A30.0084.7302  ;3726  D_R[6],SC_K[.8]    ; GET ADDRESS OF START OF MICRO CODE
U 1302. 0000.003C.0180.A800.0000.1303  ;3727  CACHE.P_D[LONG]    ; SEND IT TO CI780
U 1303. 0818.0038.D9F8.0800.0000.1304  ;3728  D_K[.9],Q_0    ; GENERATE OFFSET OF 'PICR' REG
U 1304. 0D00.003C.0180.0800.0000.1305  ;3729  D_DAL.SC    ; ...
U 1305. 0819.0030.E980.0800.0000.1306  ;3730  D_D.OR.K[.24]    ; ...
U 1306. 0011.0030.0180.0900.0200.1307  ;3731  VA_ALU,ALU_D[OR]RC[0]  ; SET ADDRESS OF 'PICR' REG
U 1307. 0818.0038.0580.0800.0000.1308  ;3732  D_K[.1]
U 1308. 0000.003C.0180.A800.0000.1116  ;3733  CACHE.P_D[LONG]    ; START THE CI780 READ
            ;3734 =0 D_K[.FF],    ; WAIT FOR APPROXIMATELY 100 MICRO-SEC
U 1116. 0818.0039.4980.0800.0000.1205  ;3735   CALL[WAIT.LOOP]
U 1117. 0000.003C.0180.0800.0000.1118  ;3736  NOP
U 1118. 0000.003D.0180.0800.0000.1249  ;3737 =0 CALL[MS780E.CLEANUP]  ; CLEANUP MEMORY
```

```
            ;3738 ;
            ;3739 ; CHECK THAT BIT 16 IN CONFIG REG A SET
            ;3740 ;
U 1119. 0010.0038.6580.0900.0284.7309  ;3741  VA_ALU,ALU_RC[0],SC_K[.10] ; LOAD REGISTER CI780 BASE ADDRESS
U 1309. 0000.003C.0180.C800.0000.130A  ;3742  D[LONG]_CACHE.P    ; READ CONFIG REG
            ;3743  RC[0D]_ALU,ALU_D.AND.MASK, ; SEE IF CRD BIT IS SET
U 130A. 0001.0034.0180.09E8.0010.130B  ;3744       CLK.UBCC    ; IS IT SET?
U 130B. 0003.011C.0180.09F0.0000.111A  ;3745  Z?,RC[0E]_NOT.MASK
U 111A. 0000.003C.0180.0800.0000.111D  ;3746 =0 J/CI780_CRD_CLR    ; BRANCH IF YES
U 111B. 0000.003C.0180.0800.0000.111C  ;3747  NOP
U 111C. 0000.003D.0180.0800.0000.116C  ;3748 =0 ERROR2     ; NO
            ;3749 ;
            ;3750 ; CHECK THAT BIT 16 IN CONFIG REG A CLEARS
            ;3751 ;
            ;3752 CI780_CRD_CLR:
U 111D. 0010.0038.6580.0900.0284.730C  ;3753  VA_ALU,ALU_RC[0],SC_K[.10] ; RELOAD CONFIG REG A ADDRESS
U 130C. 0803.001C.01C0.0800.0000.130D  ;3754  D_NOT.MASK,Q_ALU   ; GET DATA TO TRY AND CLEAR THE BIT
U 130D. 0000.003C.0180.A800.0000.130E  ;3755  CACHE.P_D[LONG]    ; TRY AND CLEAR THE BIT
U 130E. 0000.003C.0180.C800.0000.130F  ;3756  D[LONG]_CACHE.P    ; READ THE BIT
U 130F. 001D.0034.0180.09E8.0010.1310  ;3757  ALU_D.AND.Q,RC[0D]_ALU,CLK.UBCC ; DID BIT CLEAR?
U 1310. 0003.013C.0180.09F0.0000.111E  ;3758  Z?,RC[0E]_0
U 111E. 0000.003D.0180.0800.0000.116C  ;3759 =0 ERROR2     ; CRD BIT DID NOT CLEAR IN CONFIG REG
            ;3760
            ;3761 NEXUS_TEST_6:
U 111F. 0000.003C.0180.0800.0000.1120  ;3762  NOP
            ;3763 ;////////////////////////////////////////////////////////////////////////////////
            ;3764 ;
            ;3765 ; SUBTEST 6
            ;3766 ;
            ;3767 ; DW780 CRD
            ;3768 ;
            ;3769 ; THIS SUBTEST CHECKS THE CRD BIT IN THE UNIBUS ADAPTER. THIS IS DONE
            ;3770 ; BE PERFORMING A READ TO THE UNIBUS THAT GETS MAPPED INTO A READ TO
            ;3771 ; MAIN MEMORY. THE MEMORY WILL HAVE THE CRD ERROR FORCED.
            ;3772 ;
            ;3773 ; THE READ IS PERFORMED TO UNIBUS ADDRESS 20(X) WHICH MAPS THRU MAP REGISTER 0.
            ;3774 ; MAP REGISTER 0 WILL CONTAIN 80000010(X) TO GENERATE AN SBI ADDRESS OF
            ;3775 ; 4020(X).
            ;3776 ;
            ;3777 ; DATA: EXPECTED DW780 STATUS REGISTER (BIT 8 ONLY)
            ;3778 ; RECEIVED DW780 STATUS REGISTER (BIT 8 ONLY)
            ;3779 ;   SEE ABOVE
            ;3780 ;       .
            ;3781 ;       .
            ;3782 ;
U 1120. 0000.003D.0180.0800.0000.1187  ;3783 =0 SUBTEST
U 1121. 0010.0038.0180.0960.0010.1311  ;3784  ALU_RC[0C],CLK.UBCC   ; IS THIS DEVICE A DW780?
U 1311. 0000.013C.0180.0800.0000.1122  ;3785  Z?
U 1122. 0000.003C.0180.0800.0000.1137  ;3786 =0 J/NEXUS_TEST_7    ; BRANCH IF NO
U 1123. 0000.003C.0180.0800.0000.1124  ;3787  NOP
U 1124. 0000.003D.0180.0800.0000.1169  ;3788 =0 ERLOOP
U 1125. 0000.003C.0180.0800.0000.1126  ;3789  NOP
U 1126. 0000.003D.0180.0800.0000.11DD  ;3790 =0 CALL[INIT]    ; CLEANUP THE CPU AND ADAPTER
U 1127. 0000.003C.0180.0800.0000.1128  ;3791  NOP
U 1128. 0000.003D.0180.0800.0000.1191  ;3792 =0 CALL[INIT_DW780]  ;INIT THE UBA
```

```
U 1129, 0000.003C.0180.0800.0000.112A  ;3793  NOP
U 112A, 0000.003D.0180.0800.0000.11F9  ;3794 =0 CALL[WAITFORUNIBUS]  ; WAIT FOR UNIBUS TO COME UP
U 112B, 0000.003C.0180.0800.0000.112C  ;3795  NOP
U 112C, 0000.003D.0180.0800.0000.10A0  ;3796 =0 CALL[SETUP_MS780_CRD]  ; FORCE THE ERROR IN THE MEMORY
U 112D, 0000.003C.0180.0800.0000.112E  ;3797  NOP
U 112E, 0000.003D.0180.0800.0000.119F  ;3798 =0 CALL[DW780_READ]  ; CAUSE DW780 TO PERFORM THE READ
U 112F, 0000.003C.0180.0800.0000.1130  ;3799  NOP
U 1130, 0000.003D.0180.0800.0000.1249  ;3800 =0 CALL[MS780E.CLEANUP]  ; CLEANUP THE MEMORY
U 1131, 0810.0038.0180.0900.0000.1312  ;3801  D_RC[0]
U 1312, 0019.0030.0180.0800.0200.1313  ;3802  VA_ALU,ALU_D.OR.K[.8]  ; LOAD ADDRESS OF UNIBUS STATUS REG
U 1313, 0000.003C.0180.C800.0084.7314  ;3803  D[LONG]_CACHE.P,SC_K[.8] ; GET THE STATUS REGISTER
          ;3804  ALU_D.ANDNOT.MASK,RC[0D]_ALU,
U 1314, 0001.0024.0180.09E8.0010.1315  ;3805       CLK.UBCC  ; IS CRD BIT SET?
U 1315, 0003.011C.0180.09F0.0000.1132  ;3806  Z?,RC[0E]_NOT.MASK
U 1132, 0000.003C.0180.0800.0000.1135  ;3807 =0 J/DW780_CRD_CLR  ; BRANCH IF YES
U 1133, 0000.003C.0180.0800.0000.1134  ;3808  NOP
U 1134, 0000.003D.0180.0800.0000.116C  ;3809 =0 ERROR2  ; CRD BIT NOT SET
          ;3810 DW780_CRD_CLR:
U 1135, 0810.0038.0180.0900.0084.7316  ;3811  D_RC[0],SC_K[.8]
U 1316, 0019.0030.0180.0800.0200.1317  ;3812  VA_ALU,ALU_D.OR.K[.8]  ; LOAD ADDRESS OF UBA STATUS REG
U 1317, 0803.001C.0180.0800.0000.1318  ;3813  D_NOT.MASK
U 1318, 0000.003C.0180.A800.0000.1319  ;3814  CACHE.P_D[LONG]  ; TRY AND CLEAR CRD BIT
U 1319, 0000.003C.0180.C800.0000.131A  ;3815  D[LONG]_CACHE.P  ; READ THE STATUS REGISTER
          ;3816  ALU_D.ANDNOT.MASK,RC[0D]_ALU,
U 131A, 0001.0024.0180.09E8.0010.131B  ;3817       CLK.UBCC  ; DID CRD BIT CLEAR?
U 131B, 0003.013C.0180.09F0.0000.1136  ;3818  Z?,RC[0E]_0
U 1136, 0000.003D.0180.0800.0000.116C  ;3819 =0 ERROR2  ; NO
          ;3820 NEXUS_TEST_7:
U 1137, 0000.003C.0180.0800.0000.1138  ;3821  NOP
          ;3822 ;////////////////////////////////////////////////////////////////////////////
          ;3823 ;
          ;3824 ; SUBTEST 7
          ;3825 ;
          ;3826 ; RH780 CRD TEST
          ;3827 ;
          ;3828 ; THIS TEST CHECKS THE CRD BIT IN THE RH780 STATUS REGISTER
          ;3829 ;
          ;3830 ; DATA: EXPECTED RH780 STATUS REGISTER (BIT 29 ONLY)
          ;3831 ; RECEIVED RH780 STATUS REGISTER (BIT 29 ONLY)
          ;3832 ;
U 1138, 0000.003D.0180.0800.0000.1187  ;3833 =0 SUBTEST
U 1139, 0810.0038.0180.0960.0000.131C  ;3834  D_RC[0C]  ; GET DEVICE CODE
U 131C, 0019.0000.0580.0800.0010.131D  ;3835  ALU_D-K[.1],CLK.UBCC  ; RH780?
U 131D, 0000.013C.0180.0800.0000.113A  ;3836  Z?
U 113A, 0000.003C.0180.0800.0000.114B  ;3837 =0 J/NEXUS_TEST_8  ; NO
U 113B, 0000.003C.0180.0800.0000.113C  ;3838  NOP
U 113C, 0000.003D.0180.0800.0000.1169  ;3839 =0 ERLOOP
U 113D, 0000.003C.0180.0800.0000.113E  ;3840  NOP
U 113E, 0000.003D.0180.0800.0000.11DD  ;3841 =0 CALL[INIT]  ; INIT RH780
U 113F, 0000.003C.0180.0800.0000.1140  ;3842  NOP
U 1140, 0000.003D.0180.0800.0000.10A0  ;3843 =0 CALL[SETUP_MS780_CRD]  ; SETUP THE MEMORY TO FORCE THE ERROR
U 1141, 0000.003C.0180.0800.0000.1142  ;3844  NOP
U 1142, 0000.003D.0180.0800.0000.104C  ;3845 =0 CALL[RH780_READ]  ; CAUSE RH780 TO PERFORM THE READ
U 1143, 0000.003C.0180.0800.0000.1144  ;3846  NOP
U 1144, 0000.003D.0180.0800.0000.1249  ;3847 =0 CALL[MS780E.CLEANUP]  ; CLEANUP THE MEMORY
```

```
U 1145. 0010,0038,01C0,0900,0000,131E  ;3848  Q_RC[0]     ; GET RH780 BASE ADDRESS
U 131E. 0019,2030,0180,0800,0200,131F  ;3849  VA_ALU,ALU_Q.OR.K[.8]  ; SET ADDRESS OF STATUS REG
U 131F. 0000,003C,0180,C800,0000,1320  ;3850  D[LONG]_CACHE.P    ; READ STATUS REG
U 1320. 0000,003C,5180,0800,0084,7321  ;3851  SC_K[.1E]    ; GENERATE EXPECTED DATA
U 1321. 0000,003C,0580,0800,0084,B322  ;3852  SC_SC-1     ; ...
U 1322. 0801,0024,0180,0800,0010,1323  ;3853  D_D.ANDNOT.MASK,CLK.UBCC ; DID CRD BIT SET?
U 1323. 0003,011C,0180,09F0,0000,1146  ;3854  Z?,RC[0E]_NOT.MASK
U 1146. 0000,003C,0180,0800,0000,1149  ;3855 =0 J/RH780_CRD_CLR
U 1147. 0000,003C,0180,0800,0000,1148  ;3856  NOP
U 1148. 0001,003D,0180,09E8,0000,116C  ;3857 =0 ERROR2,RC[0D]_D    ; CRD BIT DID NOT SET
            ;3858 RH780_CRD_CLR:
U 1149. 0010,0038,01C0,0900,0000,1324  ;3859  Q_RC[0]
U 1324. 0019,2030,0180,0800,0200,1325  ;3860  VA_ALU,ALU_Q.OR.K[.8]  ; SET ADDRESS OF STATUS REG
U 1325. 0000,003C,5180,0800,0084,7326  ;3861  SC_K[.1E]
U 1326. 0000,003C,0580,0800,0084,B327  ;3862  SC_SC-1
U 1327. 0803,001C,0180,0800,0000,1328  ;3863  D_NOT.MASK
U 1328. 0000,003C,0180,A800,0000,1329  ;3864  CACHE.P_D[LONG]    ; TRY AND CLEAR CRD BIT
U 1329. 0000,003C,0180,C800,0000,132A  ;3865  D[LONG]_CACHE.P    ; READ STATUS REG AGAIN
U 132A. 0801,0024,0180,0800,0010,132B  ;3866  D_D.ANDNOT.MASK,CLK.UBCC ; DID BIT CLEAR?
U 132B. 0003,013C,0180,09F0,0000,114A  ;3867  Z?,RC[0E]_0
U 114A. 0001,003D,0180,09E8,0000,116C  ;3868 =0 ERROR2,RC[0D]_D    ; CRD BIT DID NOT CLEAR
            ;3869 NEXUS_TEST_8:
U 114B. 0000,003C,0180,0800,0000,132C  ;3870  NOP
            ;3871 CLN900: ;=================================================================
            ;3872  ;*
            ;3873  ; CHECK FOR LAST TR LEVEL (RA 0 = ^X10)
            ;3874  ; ADJUST ADDRESS RC 0,1,2
            ;3875  ;-
            ;3876  ;=================================================================
            ;3877  ;
U 132C. 0810,0038,0180,0958,0000,132D  ;3878  D_RC[0B]    ;GET CURRENT LEVEL OF TR
U 132D. 0019,0020,6580,0800,0010,132E  ;3879  ALU_D.XOR.K[.10],CLK.UBCC ; IS IT = 16
U 132E. 0000,013C,0180,0800,0000,114C  ;3880  Z?
U 114C. 0000,003C,0180,0800,0000,132F  ;3881 =0 J/CLN910    ;NO,FIND NEW TR, AND ADDRESSES
U 114D. 0000,003C,0180,0800,0000,1336  ;3882  J/CLN999    ;YES END TEST
U 132F. 0819,0014,0580,09D8,0000,1330  ;3883 CLN910: RC[0B]_ALU,D_D+K[.1)  ;ADJUST TO NEXT LEVEL
U 1330. 0818,0038,4580,0800,0000,1331  ;3884  D_K[.8000]
U 1331. 0200,003C,0180,0800,0000,1332  ;3885  D_D.RIGHT2    ; GET ADDRESS AJUSTING CONSTANT
U 1332. 0010,0038,01C0,0900,0000,1333  ;3886  Q_RC[0]     ;ADJUST RC 0
U 1333. 081D,0014,0180,0980,0000,1334  ;3887  RC[0]_ALU,D_D+Q
U 1334. 0819,0014,1180,0988,0000,1335  ;3888  RC[1]_ALU,D_D+K[.4]  ;ADJUST RC 1
U 1335. 0819,0014,1180,0990,0000,10A8  ;3889  RC[2]_ALU,D_D+K[.4],J/CLN000 ;ADJUST RC 2,REPEAT LOOP
U 1336. 0000,003C,0180,0800,0000,114E  ;3890 CLN999: NOP     ;END TEST
            ;3891
```

```
            ;3892  .PAGE "TEST 214 RESERVED"
            ;3893 =0
U 114E, 0000,003D,0180,0800,0000,1109  ;3894 T214: NEWTST
U 114F, 0000,003C,0180,0800,0000,1150  ;3895  NOP
            ;3896
```

```
            ;3897  .PAGE "TEST 215 RESERVED"
            ;3898 =0
U 1150, 0000,003D,0180,0800,0000,1109  ;3899 T215: NEWTST
U 1151, 0000,003C,0180,0800,0000,1152  ;3900  NOP
            ;3901
```

```
            ;3902  .PAGE "TEST 216 RESERVED"
            ;3903 =0
U 1152. 0000.003D.0180.0800.0000.1109  ;3904 T216: NEWTST
U 1153. 0000.003C.0180.0800.0000.1154  ;3905  NOP
            ;3906
```

```
          ;3907  .PAGE "DUMMY NEWTST"
          ;3908 =0
U 1154, 0000,003D,0180,0800,0000,1109  ;3909 ENDTST: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2181
ESKAR57.MCR                    MICRO2  1P(00)     4-NOV-85  13:15:47     Page  10
:                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1917: 1886 1887 3442= 1888= 1907= 1910 3444=
U 1008   2629= 2630= 2631= 2632= 2633= 2634= 2635= 2636=
U 1010   2874= 2875= 2876= 1911 1908= 1909= 1912 3359=
U 1018   2881= 2882= 2883= 1913 1925= 1926= 1914 3360=
U 1020   3363= 3364= 3365= 3366= 3367= 3368= 3369= 3370=
U 1028   3371= 3372= 3373= 3374= 3375= 3376= 3377= 3378=
U 1030   2887= 2888= 2889= 1915 2893= 2896= 2898= 2899=
U 1038   3057= 3058= 3059= 3061= 3106= 3107= 3108= 3110=
U 1040   3350= 3351= 3352= 3353= 1980= 1982= 1993= 1994=
U 1048   2015= 2016= 2040= 2041= 2272= 2273= 2022: 1916
U 1050   2294= 2295= 2300= 2301= 2306= 2307= 2310= 2311=
U 1058   2315= 2316= 2047: 1923 2343= 2344= 2051: 1924
U 1060   2355= 2356= 2369= 2370= 2376= 2377= 2382= 2383=
U 1068   2393= 2394= 2395= 2397= 2398= 2399= 2400= 2401=
U 1070   2402= 2403= 2404= 2405= 2406= 2407= 2409= 2410=
U 1078   2489= 2490= 2534= 2535= 2568= 2569= 2608= 2609=
U 1080   2617= 2618= 2619= 2621= 2624= 2625= 2666= 2667=
U 1088   2668= 2669= 2676= 2677= 2683= 2685= 2688= 2689=
U 1090   2709= 2710= 2712= 2713= 2807= 2808= 2811= 2812=
U 1098   2823= 2824= 2827= 2828= 3011= 3012= 3155= 3157=
U 10A0   3219= 3221= 3301= 3302= 3303= 3304= 3305= 3307=
U 10A8   3338= 3339= 3340= 3341= 3347= 3348= 3410= 3414=
U 10B0   3418= 3419= 3451= 3452= 3475= 3476= 3496= 3497=
U 10B8   3500= 3501= 3506= 3507= 3508= 3509= 3510= 3511=
U 10C0   3518= 3522= 3536= 3537= 3538= 3539= 3545= 3546=
U 10C8   3547= 3552= 3557= 3558= 3559= 3564= 3571= 3573=
U 10D0   3593= 3594= 3596= 3597= 3598= 3599= 3600= 3601=
U 10D8   3602= 3603= 3604= 3605= 3606= 3607= 3608= 3609=
U 10E0   3610= 3611= 3617= 3618= 3619= 3621= 3629= 3632=
U 10E8   3646= 3647= 3650= 3651= 3652= 3653= 3654= 3655=
U 10F0   3656= 3657= 3658= 3659= 3660= 3661= 3666= 3667=
U 10F8   3668= 3670= 3677= 3679= 3698= 3699= 3702= 3703=
U 1100   1872: 1873: 1874: 1875: 1876: 1877: 1878: 1879:
U 1108   1880: 1969 3708= 3709= 1881: 1882: 1883: 1884:
U 1110   3710= 3711= 3712= 3713= 3720= 3724= 3735= 3736=
U 1118   3737= 3741= 3746= 3747= 3748= 3753= 3759= 3762=
U 1120   3783= 3784= 3786= 3787= 3788= 3789= 3790= 3791=
U 1128   3792= 3793= 3794= 3795= 3796= 3797= 3798= 3799=
U 1130   3800= 3801= 3807= 3808= 3809= 3811= 3819= 3821=
U 1138   3833= 3834= 3837= 3838= 3839= 3840= 3841= 3842=
U 1140   3843= 3844= 3845= 3846= 3847= 3848= 3855= 3856=
U 1148   3857= 3859= 3868= 3870= 3881= 3882= 3894= 3895=
U 1150   3899= 3900= 3904= 3905= 3909= 1970 1971 1972
U 1158   1973 1974 1975 1976 1977 1978 1979 1983
U 1160   1984 1985 1986 1987 1988 1989 1990 1991
U 1168   1992 1996 2020 2021 2045 2046 2070 2071
U 1170   2072 2073 2111 2112 2113 2114 2115 2116
U 1178   2117 2118 2119 2120 2121 2122 2123 2124
U 1180   2125 2126 2128 2129 2130 2131 2133 2143
U 1188   2144 2167 2194 2195 2196 2205 2206 2207
U 1190   2212 2219 2220 2221 2222 2229 2230 2231
U 1198   2232 2233 2234 2240 2241 2242 2243 2252
```

ZZ-ESKAR-V22 Line Number Index
ESKAR57.MCR MICRO2 1P(00) 4-NOV-85 13:15:47 SEQ 2182 Page 10

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2253 2255 2256 2257 2258 2259 2260 2261
U 11A8  2262 2263 2274 2276 2277 2278 2279 2280
U 11B0  2281 2282 2283 2284 2285 2286 2287 2288
U 11B8  2289 2290 2291 2292 2296 2297 2298 2302
U 11C0  2303 2304 2308 2312 2313 2317 2337 2338
U 11C8  2339 2340 2341 2342 2345 2346 2347 2348
U 11D0  2349 2350 2351 2352 2353 2354 2365 2367
U 11D8  2368 2374 2375 2380 2381 2392 2411 2412
U 11E0  2421 2422 2423 2431 2432 2433 2434 2442
U 11E8  2443 2444 2445 2452 2453 2454 2455 2456
U 11F0  2457 2465 2466 2467 2468 2475 2476 2477
U 11F8  2478 2485 2486 2487 2488 2509 2510 2511
U 1200  2512 2536 2537 2538 2539 2563 2610 2611
U 1208  2612 2613 2622 2623 2627 2628 2641 2642
U 1210  2644 2645 2646 2648 2653 2654 2655 2657
U 1218  2662 2664 2665 2673 2674 2678 2679 2680
U 1220  2681 2682 2690 2691 2692 2694 2695 2696
U 1228  2697 2698 2699 2705 2707 2708 2711 2736
U 1230  2737 2738 2739 2769 2770 2772 2773 2774
U 1238  2803 2805 2806 2810 2814 2815 2816 2817
U 1240  2819 2820 2821 2822 2825 2826 2872 2873
U 1248  2900 2948 2949 2951 2952 2953 2955 2957
U 1250  2958 2960 2962 2963 2964 2965 2968 2969
U 1258  2970 2973 3000 3001 3002 3003 3004 3005
U 1260  3006 3007 3009 3010 3158 3159 3161 3162
U 1268  3164 3165 3166 3169 3170 3171 3172 3173
U 1270  3174 3222 3223 3225 3226 3228 3229 3230
U 1278  3233 3234 3235 3236 3237 3238 3308 3309
U 1280  3311 3312 3313 3314 3315 3316 3317 3318
U 1288  3319 3320 3342 3343 3344 3345 3346 3354
U 1290  3355 3356 3357 3358 3361 3362 3415 3416
U 1298  3417 3420 3421 3422 3423 3424 3426 3427
U 12A0  3428 3429 3430 3431 3432 3433 3434 3435
U 12A8  3436 3437 2208: 3438 3439 3440 3441 3445
U 12B0  3446 3447 3448 3456 3457 3458 3459 3460
U 12B8  3461 3462 3463 3464 3469 3471 3474 3498
U 12C0  3499 3512 3513 3514 3515 3516 3517 3523
U 12C8  3524 3525 3526 3527 3528 3529 3530 3531
U 12D0  3532 3533 3534 3540 3541 3542 3543 3544
U 12D8  3553 3555 3556 3565 3566 3567 3568 3569
U 12E0  3570 3595 3612 3613 3615 3616 3622 3623
U 12E8  3624 3625 3627 3628 3648 3649 3662 3663
U 12F0  3664 3665 3671 3672 3673 3674 3675 3676
U 12F8  3700 3701 3714 3715 3716 3717 3718 3719
U 1300  3725 3726 3727 3728 3729 3730 3731 3732
U 1308  3733 3742 3744 3745 3754 3755 3756 3757
U 1310  3758 3785 3802 3803 3805 3806 3812 3813
U 1318  3814 3815 3817 3818 3835 3836 3849 3850
U 1320  3851 3852 3853 3854 3860 3861 3862 3863
U 1328  3864 3865 3866 3867 3878 3879 3880 3883
U 1330  3884 3885 3886 3887 3888 3889 3890
U 1338 - 13EF Unused
U 13F0  2079: 2080: 2081: 2082: 2083: 2084: 2085: 2086:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2183
ESKAR57.MCR                    MICRO2  1P(00)     4-NOV-85  13:15:47     Page  10

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 13F8  2087: 2088: 2089: 2090: 2091: 2092: 2093: 2094:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR57	839
Used	839
Remaining	

Total microwords used in memory U: 839
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2185
ESKAR57.MCR                    MICRO2  1P(00)     4-NOV-85  13:15:47     Page  10


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR57.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2186
ESKAR58.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1933 Micro Monitor Interface Routines
; 1934   Newtest Routine
; 1992   ERROR1 WITH PARAMETER
; 2017   ERROR 2 WITH PARAMETER
; 2067   RESET SUBTEST NUMBER
; 2090   Set Trap Mask
; 2119   Disable Cache Routine
; 2206   Wait Loop Routine
; 2274   Initialize the SBI
; 2489   Find MS780-E Memory
; 2632   Generate IO Address
; 2658   Set Starting Address
; 2693   CONFIGURE MS780-E/H
; 2767   ENABLE NEXT MS780-E
; 2820   Select Controller
; 2859   SELECT LOWER CONTROLLER
; 2907   SELECT UPPER CONTROLLER
; 2956 TEST 217 NEXUS SBI FAULT TEST
; 4055 TEST 218 RESERVED
; 4060 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR58
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843 ;*
            ;1844 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1845 ; WITH A NUMBER, AND LOADS THE Q REGISTER WITH THE CONTENTS OF R[0F].
            ;1846 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1847 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1848 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1849 ;
            ;1850 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1851 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1852 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1853 ; R[0F] AND DO A RETURN0.
            ;1854 ;
            ;1855 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1856 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1857 ; TO THE CONSOLE.
            ;1858 ;
            ;1859 ; THE JUMP FIELDS OF ALL THE TRAP VECTORS EXCEPT THE CONTROL STORE PARITY
            ;1860 ; ERROR VECTOR, POINT TO A ROUTINE THAT EXECUTES A RETURN0 IF THE TRAP WAS
            ;1861 ; EXPECTED. THE CS PARITY ERROR VECTOR POINTS TO A ROUTINE THAT EXECUTES A
            ;1862 ; RETURN1 IF THE TRAP WAS EXPECTED SINCE THE ADDRESS THAT WAS PUSHED ON THE
            ;1863 ; STACK IS THE ADDRESS OF THE MICRO WORD WITH BAD PARITY.
            ;1864 ;-
            ;1865
U 1100, 0000,003C,19C0,0A78,0084,7001   ;1866 1100: SC_K[ZERO],Q_R[0F],J/TRPH0 ; SYSTEM INIT
U 1101, 0000,003C,05C0,0A78,0084,7001   ;1867 1101: SC_K[.1],Q_R[0F],J/TRPH0 ; DATA UNALIGNED
U 1102, 0000,003C,09C0,0A78,0084,7001   ;1868 1102: SC_K[.2],Q_R[0F],J/TRPH0 ; X PAGE ERROR
U 1103, 0000,003C,0DC0,0A78,0084,7001   ;1869 1103: SC_K[.3],Q_R[0F],J/TRPH0 ; TB MODIFY
U 1104, 0000,003C,11C0,0A78,0084,7001   ;1870 1104: SC_K[.4],Q_R[0F],J/TRPH0 ; TB PROT ERROR
U 1105, 0000,003C,D5C0,0A78,0084,7001   ;1871 1105: SC_K[.6],Q_R[0F],J/TRPH0 ; TB MISS
U 1106, 0000,003C,D9C0,0A78,0084,7001   ;1872 1106: SC_K[.9],Q_R[0F],J/TRPH0 ; RES FLOAT OPER
U 1107, 0000,003C,5DC0,0A78,0084,7001   ;1873 1107: SC_K[.7],Q_R[0F],J/TRPH0 ; TB PARITY
U 1108, 0000,003C,01C0,0A78,0084,7001   ;1874 1108: SC_K[.8],Q_R[0F],J/TRPH0 ; CACHE PARITY
U 110C, 0000,003C,85C0,0A78,0084,7001   ;1875 110C: SC_K[.C],Q_R[0F],J/TRPH0 ; RDS
U 110D, 0000,003C,89C0,0A78,0084,7001   ;1876 110D: SC_K[.D],Q_R[0F],J/TRPH0 ; TIME OUT
U 110E, 0000,003C,65C0,0A78,0084,7001   ;1877 110E: SC_K[.10],Q_R[0F],J/TRPH0 ; ODD ADDRESS
U 110F, 0000,003C,61C0,0A78,0084,705B   ;1878 110F: SC_K[.F],Q_R[0F],J/TRPH2 ; CS PARITY
            ;1879
U 1001, 0001,2024,0180,0800,0010,1007   ;1880 TRPH0: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1007, 0000,013C,0180,0800,0000,1002   ;1881  Z?     ; BRANCH IF NO
U 1002, 0001,2036,0180,0AF8,0000,0000   ;1882 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1883
            ;1884
            ;1885 ;*
            ;1886 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1887 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1888 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1889 ;
            ;1890 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1891 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1892 ;       FAULT STATUS REGISTER
            ;1893 ;       SBI ERROR REGISTER
            ;1894 ;       TIMEOUT ADDRESS REGISTER
            ;1895 ;       MAINTENANCE REGISTER
```

```
                ;1896 ;       CACHE PARITY REGISTER
                ;1897 ;       TB ERROR REGISTER 1
                ;1898 ;       TB ERROR REGISTER 0
                ;1899 ;-
                ;1900
U 1003. 0018.0038.1D80.09E8.0000.1014  ;1901 TRPH1: RC[0D]_SC
U 1014. 0000.003D.D980.0B00.0084.71DA  ;1902 =0 SETRCF[.9]
U 1015. 0000.003C.49F0.2C00.0000.1013  ;1903  Q_ID[TBER0]
U 1013. 0001.203C.4DF0.2DB0.0000.1016  ;1904  RC[6]_Q,Q_ID[TBER1]
U 1016. 0001.203C.65F0.2DB8.0000.101B  ;1905  RC[7]_Q,Q_ID[SBI.ERR]
U 101B. 0001.203C.6DF0.2DD8.0000.101E  ;1906  RC[0B]_Q,Q_ID[FAULT]
U 101E. 0001.203C.75F0.2DE0.0000.104F  ;1907  RC[0C]_Q,Q_ID[MAINT]
U 104F. 0001.203C.79F0.2DC8.0000.1053  ;1908  RC[9]_Q,Q_ID[PARITY]
U 1053. 0001.203C.69F0.2DC0.0000.1057  ;1909  RC[8]_Q,Q_ID[TIME.ADDR]
U 1057. 0001.203C.81F0.2DD0.0000.1000  ;1910  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.11B7  ;1911 1000: RC[0E]_Q,ERROR1
                ;1912
                ;1913 ;*
                ;1914 ; THIS ROUTINE IS USED FOR CONTROL STORE PARITY ERROR MICRO TRAPS.
                ;1915 ;-
                ;1916
U 105B. 0001.2024.0180.0800.0010.105F  ;1917 TRPH2: ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS THE TRAP EXPECTED?
U 105F. 0000.013C.0180.0800.0000.101C  ;1918  Z?     ; BRANCH IF NO
U 101C. 0001.2036.0180.0AF8.0000.0001  ;1919 =0 ALU_Q[AND]MASK,R[0F]_ALU,RETURN1 ; RETURN
U 101D. 0000.003C.0180.0800.0000.1003  ;1920  J/TRPH1     ; REPORT UNEXPECTED TRAP
                ;1921
                ;1922
                ;1923 ;*
                ;1924 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1925 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1926 ;  NEWTST
                ;1927 ;  ERLOOP
                ;1928 ;  ERROR1
                ;1929 ;  ERROR2
                ;1930 ;
                ;1931
```

```
            ;1932  .PAGE
            ;1933  .TOC " Micro Monitor Interface Routines"
            ;1934  .TOC "   Newtest Routine"
            ;1935 ;++
            ;1936 ; FUNCTIONAL DESCRIPTION:
            ;1937 ;
            ;1938 ; This routine Initializes the following registers:
            ;1939 ;  PSL    to 1F
            ;1940 ;  CES    to 0
            ;1941 ;  ACC.CS    to disable accellerator
            ;1942 ;  SIR    to 0
            ;1943 ;  CLK.CS    to stop interval timer
            ;1944 ;  TBER0    to disable the TB
            ;1945 ;  SBI.MAINT to     0
            ;1946 ;  SBI.ERR    to 0
            ;1947 ;  FAULT    to 0
            ;1948 ;  COMP    to 0
            ;1949 ;  RC[0E]    to  0
            ;1950 ;  R[0F]    to 0
            ;1951 ; It also performs an SBI UNJAM and disables the CACHE.
            ;1952 ; A SCOPE call is then made to the Monitor.
            ;1953 ;
            ;1954 ; CALLING CONSTRAINT:
            ;1955 ;
            ;1956 ; =0
            ;1957 ;
            ;1958 ; SIDE EFFECTS:
            ;1959 ;
            ;1960 ; D Reg is destroyed
            ;1961 ; SC is destroyed
            ;1962 ;--
U 1063, 0000,003C,65F8,0800,0084,7109  ;1963 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1109, 0818,0038,8D80,0800,0000,11A3  ;1964  d_k[.1f]  ; D=1F
U 11A3, 0D00,003C,0180,0800,0000,11A4  ;1965  d_dai.sc  ; D=001F0000
U 11A4, 0000,003C,3D80,3C00,0000,11A5  ;1966  id[psl]_d  ; psl = 001F0000
U 11A5, 0818,0038,0580,0800,0000,11A6  ;1967  d_k[.1]  ; Clear the CES register
U 11A6, 0D00,003C,0180,0800,0000,11A7  ;1968  d_dai.sc  ; D = 00010000
U 11A7, 0F00,003C,3180,3C00,0000,11A8  ;1969  id[ces]_d,d_0  ; ces = 00010000
U 11A8, 0000,003C,5D80,3C00,0000,11A9  ;1970  id[acc.cs]_d  ; acc.cs = 0
U 11A9, 0000,003C,3980,3C00,0000,11AA  ;1971  id[sir]_d  ; sir = 0
U 11AA, 0000,003C,2980,3C00,0000,11AB  ;1972  id[clk.cs]_d  ; clk.cs = 0
U 11AB, 0000,003C,4980,3C00,0000,104C  ;1973  id[tber0]_d  ; tber0 = 0
U 104C, 0000,003D,0180,0800,0000,11F1  ;1974 =0 call[sbi.unjam]  ; Unjam the SBI
            ;1975  intrpt.strobe,  ; Strobe interrupts
U 104D, 0818,0038,C180,0800,4000,11AC  ;1976      d_k[.ffff]  ; Setup to clear SBI error register and
U 11AC, 0801,0028,6580,3C00,0000,11AD  ;1977  id[sbi.err]_d,d_not.d ; and fault register
U 11AD, 0F00,003C,6D80,3C00,0000,11AE  ;1978  id[fault]_d,d_0  ; ...
U 11AE, 0000,003C,6D80,3C00,0000,11AF  ;1979  id[fault]_d  ; fault = 0
U 11AF, 0000,003C,7580,3C00,0000,11B0  ;1980  id[maint]_d  ; MAINT = 0
U 11B0, 0000,003C,6580,3C00,0000,11B1  ;1981  id[sbi.err]_d  ; sbi error reg = 0
U 11B1, 0000,003C,7180,3C00,0000,11B2  ;1982  id[comp]_d  ; comp reg = 0
U 11B2, 0000,003C,E580,3C00,0000,11B3  ;1983  id[T9]_d  ; Clear code for subroutine START.TEST
U 11B3, 0001,003C,0180,09F0,0000,11B4  ;1984  rc[0e]_d  ;
U 11B4, 0001,003C,0180,0AF8,0000,11B5  ;1985  r[0f]_d  ; Clear expected trap mask
U 11B5, 0001,003C,0180,09F8,0000,1058  ;1986  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
U 1058, 0000,003D,0180,0800,0000,11BF  ;1987 =0 call[disable.cache] ; Disable the cache
U 1059, 0F03,003C,0180,09E8,0000,105E  ;1988  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
        ;1989
U 11B6, 0818,0038,0980,0800,0000,105E  ;1990 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
```

```
            ;1991  .PAGE
            ;1992  .TOC "    ERROR1 WITH PARAMETER"
            ;1993 ;**
            ;1994 ; FUNCTIONAL DESCRIPTION:
            ;1995 ;
            ;1996 ; This subroutine takes  as parameter the number of arguments
            ;1997 ; to be printed to the console in the case of an error logout.
            ;1998 ;
            ;1999 ; CALLING CONSTRAINT:
            ;2000 ;
            ;2001 ; =0
            ;2002 ;INPUTS:
            ;2003 ;
            ;2004 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2005 ;--
            ;2006 =0
            ;2007 ERR1.ARG:
U 105C, 0000,003D,0180,0800,0000,11DA  ;2008  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 105D, 0000,003C,0180,0800,0000,11B7  ;2009  J/ERROR1  ; Go to ERROR1
            ;2010 ;
            ;2011 ;
            ;2012
U 11B7, 0810,0038,0180,0978,0000,11B8  ;2013 ERROR1: D_RC[0F]
U 11B8, 0819,0034,4D80,0800,0000,104E  ;2014  D_D.AND.K[.FF00]
U 104E, 0819,0030,1180,0800,0000,105E  ;2015 104E: D_D.OR.K[.4],J/CALLCON
```

```
            ;2016  .PAGE
            ;2017  .TOC "    ERROR 2 WITH PARAMETER"
            ;2018 ;++
            ;2019 ; FUNCTIONAL DESCRIPTION:
            ;2020 ;
            ;2021 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2022 ; except that in this case after setting the number of arguments
            ;2023 ; too the console, control is transferred to routine ERROR2.
            ;2024 ;
            ;2025 ; INPUTS:
            ;2026 ;
            ;2027 ;       RC[0F] Gets the number of parameters to be printed on the console
            ;2028 ;
            ;2029 ;--
            ;2030 =0
            ;2031 ERR2.ARG:
U 1068, 0000,003D,0180,0800,0000,11DA  ;2032  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1069, 0000,003C,0180,0800,0000,11B9  ;2033  J/ERROR2  ; Go to ERROR2
            ;2034 ;
            ;2035 ;
            ;2036
U 11B9, 0810,0038,0180,0978,0000,11BA  ;2037 ERROR2: D_RC[0F]
U 11BA, 0819,0034,4D80,0800,0000,105A  ;2038  D_D.AND.K[.FF00]
U 105A, 0819,0030,D580,0800,0000,105E  ;2039 105A: D_D.OR.K[.6],J/CALLCON
            ;2040
            ;2041 105E:
            ;2042 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2043  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;2044
            ;2045 ;*
            ;2046 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;2047 ;-
            ;2048
U 13F0, 0000,003C,0180,0800,0000,13F1  ;2049 13F0: NOP
U 13F1, 0000,003C,0180,0800,0000,13F2  ;2050 13F1: NOP
U 13F2, 0000,003C,0180,0800,0000,13F3  ;2051 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;2052 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;2053 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;2054 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;2055 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;2056 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;2057 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;2058 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;2059 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;2060 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;2061 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;2062 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;2063 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,11BB  ;2064 13FF: NOP
            ;2065
```

```
            ;2066  .PAGE
            ;2067  .TOC "   RESET SUBTEST NUMBER"
            ;2068 ;++
            ;2069 ; FUNCTIONAL DESCRIPTION:
            ;2070 ;
            ;2071 ; Since some  of the tests will have to be restarted when two
            ;2072 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2073 ; the subtest counter to zero ( only for thoes tests that have
            ;2074 ; more than one subtest). This subroutine may also be necessary
            ;2075 ; when a test is being loop on.
            ;2076 ;
            ;2077 ; CALLING CONSTRAINT:
            ;2078 ;
            ;2079 ; =0
            ;2080 ; SIDE EFFECTS:
            ;2081 ;
            ;2082 ; D is destroyed
            ;2083 ;
            ;2084 ;--
            ;2085 RESET.SUBTST:
            ;2086  RC[0F]_0,    ; Reset subtest number
U 11BB, 0003,003E,0180,09F8,0000,0001  ;2087   RETURN1    ; ...and return
            ;2088
```

```
          ;2089  .PAGE
          ;2090  .TOC "   Set Trap Mask"
          ;2091 ;++
          ;2092 ; FUNCTIONAL DESCRIPTION:
          ;2093 ;
          ;2094 ; This routine is used to set a bit in the Micro Trap Mask
          ;2095 ; R[0F].
          ;2096 ;
          ;2097 ; CALLING CONSTRAINT:
          ;2098 ;
          ;2099 ; =0
          ;2100 ;
          ;2101 ; INPUTS:
          ;2102 ;
          ;2103 ; SC - Contains bit number to set.
          ;2104 ;
          ;2105 ; OUTPUTS:
          ;2106 ;
          ;2107 ; rc[0E] - Contains the current trap mask
          ;2108 ;
          ;2109 ; SIDE EFFECTS:
          ;2110 ;
          ;2111 ; d regsiter is destroyed
          ;2112 ;--
          ;2113 SET_TRAP_MASK:
U 11BC. 0800.003C.0180.0A78.0000.11BD  ;2114  d_r[0f]
U 11BD. 0801.001C.0180.0AF8.0000.11BE  ;2115  r[0f]_d.ornot.mask,d_alu ; Set mask
U 11BE. 0001.003E.0180.0AF0.0000.0001  ;2116  r[0E]_d,return1    ; Save mask and return
          ;2117
```

```
            ;2118  .PAGE
            ;2119  .TOC "   Disable Cache Routine"
            ;2120 ;++
            ;2121 ; FUNCTIONAL DESCRIPTION:
            ;2122 ;
            ;2123 ; This routine disables cache hits by setting bits <16:15>
            ;2124 ; in the maintenance register.
            ;2125 ;
            ;2126 ; CALLING SEQUENCE:
            ;2127 ;
            ;2128 ; =0
            ;2129 ;
            ;2130 ; SIDE EFFECTS:
            ;2131 ;
            ;2132 ; D & Q Registers destroyed
            ;2133 ;--
            ;2134 DISABLE.CACHE:
U 11BF. 0818.0038.4580.0800.0000.11C0  ;2135  d_k[.8000]  ; ... and seed constant
U 11C0. 0500.003C.0180.0800.0000.11C1  ;2136  d_d.left  ; Generate final constant
U 11C1. 0819.0030.4580.0800.0000.11C2  ;2137  d_d.or.k[.8000]  ; ... = 00018000
U 11C2. 0000.003E.7580.3C00.0000.0001  ;2138  id[maint]_d,return1 ; Disable cache and return
            ;2139
            ;2140 ;+
            ;2141 ; THESE SUBROUTINES ARE USED TO GENERATE CONSTANTS IN THE D REGISTER
            ;2142 ; THE D REGISTER IS SHIFTED LEFT BY 4 BITS AND THE APPROPRIATE CONSTANT
            ;2143 ; IS INSERTED INTO THE LOWER 4 BITS. THE CONSTANT THAT GETS INSERTED
            ;2144 ; IS DETERMINED BY THE ENTRY POINT INTO THE FOLLOWING 16 LOCATIONS.
            ;2145 ;
            ;2146 ; FOR EXAMPLE: IF A HEX "55" IS DESIRED IN THE LOW 8 BITS OF THE D REG THEN
            ;2147 ; THE FOLLOWING CODE WOULD BE USED TO GET THAT CONSTANT
            ;2148 ;
            ;2149 ;   =0 D_0,CALL,J/DC5
            ;2150 ;    NOP
            ;2151 ;   =0 CALL,J/DC5
            ;2152 ;-
            ;2153
U 11C3. 0018.0038.1980.0800.0000.11D3  ;2154 DC0: ALU_0(K),J/S00
U 11C4. 0018.0038.1980.0800.0000.11D4  ;2155 DC1: ALU_0(K),J/S01
U 11C5. 0018.0038.1980.0800.0000.11D5  ;2156 DC2: ALU_0(K),J/S10
U 11C6. 0018.0038.1980.0800.0000.11D6  ;2157 DC3: ALU_0(K),J/S11
U 11C7. 0018.0038.0980.0800.0000.11D3  ;2158 DC4: ALU_K[.2],J/S00
U 11C8. 0018.0038.0980.0800.0000.11D4  ;2159 DC5: ALU_K[.2],J/S01
U 11C9. 0018.0038.0980.0800.0000.11D5  ;2160 DC6: ALU_K[.2],J/S10
U 11CA. 0018.0038.0980.0800.0000.11D6  ;2161 DC7: ALU_K[.2],J/S11
U 11CB. 0018.0038.0580.0800.0000.11D3  ;2162 DC8: ALU_K[.1],J/S00
U 11CC. 0018.0038.0580.0800.0000.11D4  ;2163 DC9: ALU_K[.1],J/S01
U 11CD. 0018.0038.0580.0800.0000.11D5  ;2164 DCA: ALU_K[.1],J/S10
U 11CE. 0018.0038.0580.0800.0000.11D6  ;2165 DCB: ALU_K[.1],J/S11
U 11CF. 0018.0038.C180.0800.0000.11D3  ;2166 DCC: ALU_K[.FFFF],J/S00
U 11D0. 0018.0038.C180.0800.0000.11D4  ;2167 DCD: ALU_K[.FFFF],J/S01
U 11D1. 0018.0038.C180.0800.0000.11D5  ;2168 DCE: ALU_K[.FFFF],J/S10
U 11D2. 0018.0038.C180.0800.0000.11D6  ;2169 DCF: ALU_K[.FFFF],J/S11
            ;2170
U 11D3. 0118.0038.1B80.0800.0000.11D7  ;2171 S00: ALU_0(K),D_D.LEFT2,SI/7,J/SX
U 11D4. 0118.0038.0B80.0800.0000.11D7  ;2172 S01: ALU_K[.2],D_D.LEFT2,SI/7,J/SX
```

```
U 11D5. 0118,0038,0780,0800,0000,11D7  ;2173 S10: ALU_K[.1],D_D.LEFT2,SI/7,J/SX
U 11D6. 0118,0038,C380,0800,0000,11D7  ;2174 S11: ALU_K[.FFFF],D_D.LEFT2,SI/7,J/SX
          ;2175
U 11D7. 0100,003E,0380,0800,0000,0001  ;2176 SX: D_D.LEFT2,SI/7,RETURN1  ; EXIT
          ;2177
          ;2178
          ;2179 ;*
          ;2180 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
          ;2181 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
          ;2182 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
          ;2183 ; TYPING IT.
          ;2184 ;-
          ;2185
U 11D8. 0810,0038,4D80,0978,0000,11D9  ;2186 SUBTST: D_RC[0F],KMX/.FF00
U 11D9. 0019,4016,4D80,09F8,0000,0001  ;2187  RC[0F]_D*K[.FF00],WORD,RETURN1
          ;2188
          ;2189 ;*
          ;2190 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
          ;2191 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
          ;2192 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
          ;2193 ;-
          ;2194
U 11DA. 0010,0038,01C0,0978,0000,11DB  ;2195 SETRCF: Q_RC[0F]
U 11DB. 0019,2034,4DC0,0800,0000,11DC  ;2196  Q_Q.AND.K[.FF00]
U 11DC. 0019,2032,1D80,09F8,0000,0001  ;2197  RC[0F]_Q.OR.SC,RETURN1
U 12AA. 000D,003D,0180,0800,0000,11B7  ;2198 12AA: ERROR1     ;U-ECO LOACTION
          ;2199
          ;2200
          ;2201
U 11DD. 0818,0038,0180,0800,0000,105E  ;2202 DELAY: D_K[.8],J/CALLCON
          ;2203
          ;2204
```

```
            ;2205  .PAGE
            ;2206  .TOC "   Wait Loop Routine"
            ;2207 ;++
            ;2208 ; FUNCTIONAL DESCRIPTION:
            ;2209 ;
            ;2210 ; This routine is used to wait the specified number of machine cycles.
            ;2211 ; This routine is typically called to wait for an SBI write to
            ;2212 ; I/O space to complete.
            ;2213 ;
            ;2214 ; CALLING CONSTRAINT:
            ;2215 ;
            ;2216 ; =0
            ;2217 ;
            ;2218 ; INPUTS:
            ;2219 ;
            ;2220 ; d - Contains number of cycles to wait
            ;2221 ; alu.z condition code must not be set
            ;2222 ;
            ;2223 ; SIDE EFFECTS:
            ;2224 ;
            ;2225 ; d is destroyed
            ;2226 ;--
            ;2227 WAIT.LOOP:
U 11DE, 0018,0038,6180,0800,0010,106A  ;2228  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2229      ; ...is not set
            ;2230 =0
            ;2231 W.LOOP:
            ;2232  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 106A, 0819,0100,0580,0800,0010,106A  ;2233      j/W.LOOP
U 106B, 0000,003E,0180,0800,0000,0001  ;2234  return1   ; exit
            ;2235
            ;2236
            ;2237 NOINTR: ;=================================================================
            ;2238  ;*
            ;2239  ;
            ;2240  ; UNEXPECTED INTERRUPT ERROR LOGOUT
            ;2241  ;
            ;2242  ;DATA: INTTERRUPT VECTOR STROBED
            ;2243  ; FAULT REGISTER  ID=^X1B
            ;2244  ; SBI ERROR REGISTER ID=^X19
            ;2245  ; TIMEOUT ADDRESS  ID=^X1A
            ;2246  ; MAINT REGISTER  ID=^X1D
            ;2247  ; CACHE PARITY ERROR REG ID=^X1E
            ;2248  ; TB ERROR REG 1  ID=^X13
            ;2249  ; TB ERROR REG 0  ID=^X12
            ;2250  ; TR LEVEL UNDER TEST
            ;2251  ;-
            ;2252  ;=================================================================
            ;2253  ;
U 11DF, 0000,003C,0180,0800,4000,11E0  ;2254  IEK/ISTR  ;STROBE THE VECTOR
U 11E0, 0000,003C,0180,0800,0000,11E1  ;2255  NOP
U 11E1, 0000,003C,35F0,2C00,0000,11E2  ;2256  Q_ID[VECTOR]
U 11E2, 0F19,2034,81C0,0800,0000,11E3  ;2257  Q_Q.AND.K[.3FF],D_0
U 11E3, 001D,0020,0180,0800,0010,11E4  ;2258  ALU_D[XOR]Q,CLK.UBCC
U 11E4, 0000,013C,0180,0800,0000,106C  ;2259  Z?
```

```
U 106C. 0000.003C.0180.0800.0000.11E5   ;2260 =0 J/ERRNON  ;NOT THE EXPECTED INTERRUPT
U 106D. 0000.003E.0180.0800.0000.0002   ;2261  RETURN2    ;YES THE CORRECT ONE OCCURED
U 11E5. 0001.203C.49F0.2DF0.0000.11E6   ;2262 ERRNON: RC[0E]_Q,Q_ID[TBER0]
U 11E6. 0001.203C.4DF0.2DB8.0000.11E7   ;2263  RC[07]_Q,Q_ID[TBER1]
U 11E7. 0001.203C.79F0.2DC0.0000.11E8   ;2264  RC[08]_Q,Q_ID[PARITY]
U 11E8. 0001.203C.75F0.2DC8.0000.11E9   ;2265  RC[09]_Q,Q_ID[MAINT]
U 11E9. 0001.203C.69F0.2DD0.0000.11EA   ;2266  RC[0A]_Q,Q_ID[TIME.ADDR]
U 11EA. 0001.203C.65F0.2DD8.0000.11EB   ;2267  RC[0B]_Q,Q_ID[SBI.ERR]
U 11EB. 0001.203C.6DF0.2DE0.0000.11EC   ;2268  RC[0C]_Q,Q_ID[FAULT]
U 11EC. 0001.203C.C5F0.2DE8.0000.106E   ;2269  RC[0D]_Q,Q_ID[T1]
U 106E. 0001.203D.D980.09B0.0084.71DA   ;2270 =0 SETRCF[.9],RC[6]_Q  ;#ARG TO CONSOLE
U 106F. 0000.003E.0180.0800.0000.0001   ;2271  RETURN1
            ;2272
```

```
            ;2273  .PAGE
            ;2274  .TOC "   Initialize the SBI"
            ;2275 ;++
            ;2276 ; FUNCTIONAL DESCRIPTION:
            ;2277 ;
            ;2278 ; This routine performs the following functions:
            ;2279 ;
            ;2280 ;  Executes an SBI Unjam
            ;2281 ;  Clears the SBI Fault Latch
            ;2282 ;  Clears the SBI Error Register
            ;2283 ;
            ;2284 ; CALLING CONSTRAINT:
            ;2285 ;
            ;2286 ; =0
            ;2287 ;
            ;2288 ; SIDE EFFECTS:
            ;2289 ;
            ;2290 ; D & Q Register destroyed
            ;2291 ;--
            ;2292 =0
            ;2293 SBI.INIT:
U 1070, 0000.003D.0180.0800.0000.11F1  ;2294  call[sbi.unjam]  ; Unjam the SBI
U 1071, 0818.0038.C180.0800.0000.11ED  ;2295  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 11ED, 0801.0028.6580.3C00.0000.11EE  ;2296  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 11EE, 0F00.003C.6D80.3C00.0000.11EF  ;2297  id[fault]_d,d_0
U 11EF, 0000.003C.6D80.3C00.0000.11F0  ;2298  id[fault]_d
U 11F0, 0000.003E.6580.3C00.0000.0001  ;2299  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
            ;2300
            ;2301
            ;2302 SBI.UNJAM:
            ;2303 CLRSBI: ;=================================================================
            ;2304  ;
            ;2305  ;PERFORM AN SBI UNJAM SEQUENCE
            ;2306  ;
            ;2307  ;=================================================================
            ;2308  ;
U 11F1, 0F00.003C.0180.8000.0000.11F2  ;2309  D_0,MCT/SBI.HOLD  ;ASSERT HOLD FOR 16 CLOCK TICS
            ;2310 UNJAM0: ALU_D[XOR]K[.F],CLK.UBCC,
U 11F2, 0019.0020.6180.8000.0010.11F3  ;2311   MCT/SBI.HOLD
U 11F3, 0000.013C.0180.8000.0000.1072  ;2312  Z?,MCT/SBI.HOLD
U 1072, 0819.0014.0580.8000.0000.11F2  ;2313 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM0
U 1073, 0F00.003C.0180.8800.0000.11F4  ;2314  D_0,MCT/SBI.HOLD+UNJAM  ;ASSERT HOLD AND UNJAM FOR 16
            ;2315       ; CLOCK TICS
            ;2316
            ;2317 UNJAM1: ALU_D[XOR]K[.F],CLK.UBCC,
U 11F4, 0019.0020.6180.8800.0010.11F5  ;2318   MCT/SBI.HOLD+UNJAM
U 11F5, 0000.013C.0180.8800.0000.1074  ;2319  Z?,MCT/SBI.HOLD+UNJAM
U 1074, 0819.0014.0580.8800.0000.11F4  ;2320 =0 D_D+K[.1],MCT/SBI.HOLD+UNJAM,J/UNJAM1
U 1075, 0F00.003C.0180.8000.0000.11F6  ;2321  D_0,MCT/SBI.HOLD  ;FINISH SEQUENCE,ASSERT HOLD
            ;2322       ;  FOR THE ENDING 16 CLOCK TICS
            ;2323 UNJAM2: ALU_D[XOR]K[.F],CLK.UBCC,
U 11F6, 0019.0020.6180.8000.0010.11F7  ;2324   MCT/SBI.HOLD
U 11F7, 0000.013C.0180.8000.0000.1076  ;2325  Z?,MCT/SBI.HOLD
U 1076, 0819.0014.0580.8000.0000.11F6  ;2326 =0 D_D+K[.1],MCT/SBI.HOLD,J/UNJAM2
U 1077, 0000.003E.0180.0800.0000.0001  ;2327  RETURN1
```

```
            ;2328 INIT: ;=================================================================
            ;2329  ;
            ;2330  ;CLEARS FAULT BITS,ENABLES FAULT INTTERRUPTS,SETS
            ;2331  ;   PSL,CLEARS SOFTWARE INTERRUPT,CLEARS U-TRAP
            ;2332  ;    FLAG,UNJAMS THE SBI, SHUT CACHE AND TB OFF
            ;2333  ;
            ;2334  ;=================================================================
            ;2335  ;
U 11F8, 0003.003C.0180.0AF8.0000.1078  ;2336  R[0F]_0    ;CLEAR U-TRAPS
U 1078, 0000.003D.0180.0800.0000.11F1  ;2337 =0 CALL[CLRSBI]   ; EXECUTE AN UNJAM SEQUENCE
U 1079, 0000.003C.0180.0800.0000.107A  ;2338  NOP
U 107A, 0000.003D.0180.0800.0000.120C  ;2339 =0 CALL[SETPSL]   ;SET PSL AND CLEAR SOFTWARE
            ;2340       ;  INTRUPT REG
U 107B, 0000.003C.0180.0800.0000.107C  ;2341  NOP
U 107C, 0000.003D.0180.0800.0000.1213  ;2342 =0 CALL[CLRFLT]   ;CLEAR CPU FAULT BIT
U 107D, 0000.003C.0180.0800.0000.107E  ;2343  NOP
U 107E, 0000.003D.0180.0800.0000.121D  ;2344 =0 CALL[CLRECC]   ;CLEAR ANY PENDING ECC INTRUPTS
U 107F, 0000.003C.0180.0800.0000.1080  ;2345  NOP
U 1080, 0000.003D.0180.0800.0000.120F  ;2346 =0 CALL[CLRTIM]   ;CLEAR TIMEOUT BIT
U 1081, 0000.003C.0180.0800.0000.1082  ;2347  NOP
U 1082, 0000.003D.0180.0800.0000.1221  ;2348 =0 CALL[ENBECC]   ;ENABLE ECC INTRUPTS
U 1083, 0000.003C.0180.0800.0000.1084  ;2349  NOP
U 1084, 0000.003D.0180.0800.0000.1217  ;2350 =0 CALL[ENBFLT]   ;ENBALE FAULT INTERRUPTS
U 1085, 0818.0038.4580.0800.0000.1086  ;2351  D_K[.8000]   ;ENABLE CACHE PARITY ERROR
            ;2352 =0 CALL[DISABLE.CACHE],  ;SHUT CACHE OFF
U 1086, 0000.003D.7980.3C00.0000.11BF  ;2353   ID[PARITY]_D
U 1087, 0F00.003C.0180.0800.0000.11F9  ;2354  D_0
U 11F9, 0000.003C.4980.3C00.0000.11FA  ;2355  ID[TBER0]_D   ;SHUT TB OFF
U 11FA, 0000.003E.7180.3C00.0000.0001  ;2356  ID[COMP]_D,RETURN1  ;DISABLE SILO INTERRUPTS
            ;2357  ;
            ;2358  ;=================================================================
            ;2359  ;*
            ;2360  ;CHECK FOR INTERRUPTS PENDING
            ;2361  ;
            ;2362  ; INTERRUPT  VECTOR CHECKED
            ;2363  ; ------------------------------------
            ;2364  ; WRITE TIMEOUT  ^X60
            ;2365  ; SBI FAULT  ^X5C
            ;2366  ; CRD,RDS TAG  ^X54
            ;2367  ; SILO    ^X50
            ;2368  ;
            ;2369  ; IF THE WRONG VECTOR IS DETECTED, THE
            ;2370  ; FOLLOW ERROR LOGOUT OCCURES(ON A RETURN1)
            ;2371  ;
            ;2372  ; DATA: EXPECTED VECTOR
            ;2373  ;  VECTOR STROBED
            ;2374  ;-
            ;2375  ;=================================================================
            ;2376  ;
            ;2377 SBIFLT: ;=================================================================
            ;2378  ;CHECKS FOR THE SBI FAULT INTERRUPT, ^X5C
            ;2379  ;=================================================================
            ;2380  ;
U 11FB, 0818.0038.3580.0800.4000.11FC  ;2381  IEK/ISTR,D_K[.50]  ;STROBE THE VECTOR
U 11FC, 0819.0030.8580.0800.0000.11FD  ;2382  D_D.OR.K[.C]   ;GENERATE THE EXPECTED VECTOR
```

```
U 11FD. 0000.003C.35F0.2C00.0000.11FE  ;2383  Q_ID[VECTOR]
U 11FE. 0019.2034.81C0.0800.0000.11FF  ;2384  Q_Q.AND.K[.3FF]
U 11FF. 001D.0020.0180.0800.0010.1200  ;2385  ALU_D[XOR]Q,CLK.UBCC
U 1200. 0000.013C.0180.0800.0000.1088  ;2386  Z?
U 1088. 0000.003C.0180.0800.0000.1201  ;2387 =0 J/ERRFLT   ;NOT THE EXPECTED INTERRUPT
U 1089. 0000.003E.0180.0800.0000.0002  ;2388  RETURN2
U 1201. 0001.203C.0180.09E8.0000.108A  ;2389 ERRFLT: RC[0D]_Q   ;SET VECTOR RECIEVED TO CONSOLE
U 108A. 0001.003D.0980.09F0.0084.71DA  ;2390 =0 SETRCF[.2],RC[0E]_D  ;# OF ARG AND EXPECTED DATA
U 108B. 0000.003E.0180.0800.0000.0001  ;2391  RETURN1    ;RETURN1 TO ERROR CALL
           ;2392 WRTOUT: ;=================================================================
           ;2393  ;CHECKS FOR THE WRITE TIMEOUT INTERRUPT, ^X60
           ;2394  ;=======================================================================
           ;2395  ;
U 1202. 0818.0038.3180.0800.4000.1203  ;2396  IEK/ISTR,D_K[.40]  ;CHECK THE VECTOR
U 1203. 0000.003C.7580.0800.0000.1204  ;2397  K[.20]
U 1204. 0819.0014.7580.0800.0000.1205  ;2398  D_D+K[.20]
U 1205. 0000.003C.35F0.2C00.0000.1206  ;2399  Q_ID[VECTOR]
U 1206. 0019.2034.81C0.0800.0000.1207  ;2400  Q_Q.AND.K[.3FF]
U 1207. 0818.0038.3180.0800.0000.1208  ;2401  D_K[.40]   ;CHECK THE VECTOR
U 1208. 0000.003C.7580.0800.0000.1209  ;2402  K[.20]
U 1209. 0819.0014.7580.0800.0000.120A  ;2403  D_D+K[.20]
U 120A. 001D.2020.0180.0800.0010.120B  ;2404  ALU_Q[XOR]D,CLK.UBCC
U 120B. 0000.013C.0180.0800.0000.108C  ;2405  Z?
U 108C. 0000.003C.0180.0800.0000.1201  ;2406 =0 J/ERRFLT   ;NOT THE EXPECTED INTERRUPT
U 108D. 0000.003E.0180.0800.0000.0002  ;2407  RETURN2
           ;2408 SETPSL: ;=================================================================
           ;2409  ;
           ;2410  ;DROP THE CPU IPR LEVEL TO
           ;2411  ; ZERO, CLEAR ANY SOFTWARE INTERRUPTS
           ;2412  ; PENDING
           ;2413  ;
           ;2414  ;=======================================================================
           ;2415  ;
U 120C. 0F00.003C.0180.0800.0000.120D  ;2416  D_0
U 120D. 0000.003C.3980.3C00.0000.120E  ;2417  ID[SIR]_D  ;SET SOFTWARE LEVELS TO ZERO
U 120E. 0000.003E.3D80.3C00.0000.0001  ;2418  RETURN1,ID[PSL]_D
           ;2419 CLRTIM: ;=================================================================
           ;2420  ;
           ;2421  ;CLEAR THE CP TIMEOUT BIT IN THE
           ;2422  ; SBI ERROR REGISTER
           ;2423  ;
           ;2424  ;=======================================================================
           ;2425  ;
U 120F. 0818.0038.0580.0800.0000.1210  ;2426  D_K[.1]
U 1210. 0000.003C.85F8.0800.0084.7211  ;2427  Q_0,SC_K[.C]
U 1211. 0D00.003C.0180.0800.0000.1212  ;2428  D_DAL.SC
U 1212. 0000.003E.6580.3C00.0000.0001  ;2429  ID[SBI.ERR]_D,RETURN1
           ;2430 CLRFLT: ;=================================================================
           ;2431  ;
           ;2432  ;CLEAR THE CP FAULT BIT IN THE
           ;2433  ; FAULT REGISTER
           ;2434  ;
           ;2435  ;=======================================================================
           ;2436  ;
U 1213. 0818.0038.0180.0800.0000.1214  ;2437  D_K[.8]
```

```
U 1214, 0000.003C.65F8.0800.0084.7215  ;2438  Q_0,SC_K[.10]
U 1215, 0D00.003C.0180.0800.0000.1216  ;2439  D_DAL.SC
U 1216, 0000.003E.6D80.3C00.0000.0001  ;2440  ID[FAULT]_D,RETURN1
          ;2441 ENBFLT: ;=================================================================
          ;2442  ;
          ;2443  ;ENABLE CPU FAULT INTERRUPT
          ;2444  ;
          ;2445  ;=================================================================
          ;2446  ;
U 1217, 0818.0038.0580.0800.0000.1218  ;2447  D_K[.1]
U 1218, 0000.003C.D5F8.0800.0084.7219  ;2448  Q_0,SC_K[.6]
U 1219, 0D00.003C.0180.0800.0000.121A  ;2449  D_DAL.SC
U 121A, 0000.003C.8580.0800.0084.721B  ;2450  SC_K[.C]
U 121B, 0D00.003C.0180.0800.0000.121C  ;2451  D_DAL.SC
U 121C, 0000.003E.6D80.3C00.0000.0001  ;2452  ID[FAULT]_D,RETURN1
          ;2453 CLRECC: ;=================================================================
          ;2454  ;
          ;2455  ;CLEAR CRD,RDS BIT IN THE
          ;2456  ;SBI ERROR REGISTER
          ;2457  ;
          ;2458  ;=================================================================
          ;2459  ;
U 121D, 0818.0038.0D80.0800.0000.121E  ;2460  D_K[.3]
U 121E, 0000.003C.89F8.0800.0084.721F  ;2461  Q_0,SC_K[.D]
U 121F, 0D00.003C.0180.0800.0000.1220  ;2462  D_DAL.SC
U 1220, 0000.003E.6580.3C00.0000.0001  ;2463  ID[SBI.ERR]_D,RETURN1
          ;2464 ENBECC: ;=================================================================
          ;2465  ;
          ;2466  ;ENABLE CRD,RDS INTERRUPTS
          ;2467  ;
          ;2468  ;=================================================================
          ;2469  ;
U 1221, 0818.0038.0580.0800.0000.1222  ;2470  D_K[.1]
U 1222, 0000.003C.61F8.0800.0084.7223  ;2471  Q_0,SC_K[.F]
U 1223, 0D00.003C.0180.0800.0000.1224  ;2472  D_DAL.SC
U 1224, 0000.003E.6580.3C00.0000.0001  ;2473  ID[SBI.ERR]_D,RETURN1
          ;2474
          ;2475 ;
          ;2476 ; THIS SUBROUTINE WAITS FOR THE UNIBUS ADAPTER TO COMPLETE
          ;2477 ; AN INITIALIZATION SEQUENCE
          ;2478 ;
          ;2479 WAITFORUNIBUS:
U 1225, 0818.0038.C180.0800.0000.1226  ;2480  D_K[.FFFF]
U 1226, 0500.003C.0180.0800.0000.1227  ;2481  D_D.LEFT
U 1227, 0819.0000.0580.0800.0010.1228  ;2482 LOOPUN: D_D-K[.1],CLK.UBCC
U 1228, 0000.013C.0180.0800.0000.108E  ;2483  Z?
U 108E, 0000.003C.0180.0800.0000.1227  ;2484 =0 J/LOOPUN
U 108F, 0000.003E.0180.0800.0000.0001  ;2485  RETURN1
          ;2486
          ;2487
```

```
            ;2488  .PAGE
            ;2489  .TOC "   Find MS780-E Memory"
            ;2490 ;++
            ;2491 ; FUNCTIONAL DESCRIPTION:
            ;2492 ;
            ;2493 ;          The  diagnostic  is  designed  to  handle  up  to  4  (four)
            ;2494 ;     MS780-E/H's  on the 780 SBI.  This routine looks at all TR levels
            ;2495 ;     for a MS780-E memory unit.   Upon return, ID registers 3A through
            ;2496 ;     3D will hold the I/O base address of the MS780-E subsystems found
            ;2497 ;     on the SBI, and ID Register 3E will  hold  the  Total  number  of
            ;2498 ;     MS780-E/H's  found  minus  one.  Also, it is to be noted that the
            ;2499 ;     first MS780-E found will have  its  starting  address  set  to  0
            ;2500 ;     (zero).
            ;2501 ;     All the other MS780-E/H's found will have their starting  address
            ;2502 ;     set  to  all ones.  Subroutine ENABLE.NEXT.MS780E will look at ID
            ;2503 ;     Reg 3E to decide if there are any  more  MS780-E  and  will  take
            ;2504 ;     apropriate  action.   Register RC[0E] is used as a pointer to the
            ;2505 ;     current MS780-E unit under test.
            ;2506 ;     If any of:  MS780-A, MS780-C or MA780 is found,  its  address  is
            ;2507 ;     set to maximum.
            ;2508 ;
            ;2509 ; CALLING CONSTRAINT:
            ;2510 ;
            ;2511 ; =00
            ;2512 ;
            ;2513 ; OUTPUTS:
            ;2514 ;
            ;2515 ; rc[0e] - Contains address of Configuration Register
            ;2516 ; r[0]  - Contains TR level
            ;2517 ;
            ;2518 ; SIDE EFFECTS:
            ;2519 ;
            ;2520 ; D register is destroyed.
            ;2521 ;--
            ;2522 =0
            ;2523 FIND.MS780E:
U 1090,  0000,003D,0180,0800,0000,1070  ;2524  call[sbi.init]  ; Make sure SBI is enabled
U 1091,  0003,003C,0180,0988,0000,1229  ;2525  rc[01]_0  ; Clear "Found MS780-E/H Flag"
U 1229,  0818,0038,1980,0800,0000,122A  ;2526  d_k[zero]  ; Clear MS780-E/H counter
U 122A,  0000,003C,F980,3C00,0000,122B  ;2527  id[3e]_d  ; ...
U 122B,  0018,0038,0580,0A80,0000,122C  ;2528  r[0]_k[.1]  ; Init TR counter
U 122C,  0F03,003C,0180,09F0,0000,1092  ;2529  d_0,rc[0E]_0  ; Clear "Save" location for
            ;2530      ; ...CNFG A Reg
            ;2531 =0
            ;2532 FIND.MEM.LOOP:
U 1092,  0800,003D,0180,0A00,0000,1252  ;2533  d_r[0],call[generate.io]; Generate address of TR level
U 1093,  0001,003C,0180,09F0,0200,1094  ;2534  va_d,rc[0e]_d  ; Put address in VA and save in RC[0e]
U 1094,  0000,003D,8980,0800,0084,71BC  ;2535 =0 set.trap.mask[.d] ; Enable timeout micro traps
            ;2536  d[long]_cache.p, ; Read the specified tr level
U 1095,  0000,003C,0180,C970,0000,122D  ;2537   ic_rc[0e]  ; ...
U 122D,  0000,003C,0180,0A78,0010,122E  ;2538  alu_r[0f],clk.ubcc ; Check for no response
U 122E,  0001,013C,0180,0880,0082,1096  ;2539  z?,sc_d,ia_ra[0] ; Branch if no adapter here
U 1096,  0000,003C,0180,0800,0000,122F  ;2540 =0 j/check.adpt.code ; Check for a memory
U 1097,  0000,003C,0180,0800,0000,124E  ;2541  j/find.mem.lp1  ; Try next tr
            ;2542 CHECK.ADPT.CODE:
```

```
U 122F, 0000,003C,1D80,0800,1404,7230  ;2543  state_sc.via.kmx ; Setup to branch on bits<6:4>
U 1230, 0000,163C,0180,0800,0000,1008  ;2544  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,8980,0800,0084,7231  ;2545 =1000 j/ms780.a,sc_k[.d] ; MS780-A
U 1009, 0000,003C,8980,0800,0084,7232  ;2546  j/ms780.c,sc_k[.d] ; MS780-C
U 100A, 0000,003C,0180,0800,0000,124E  ;2547  j/find.mem.lp1  ; Not a memory
U 100B, 0000,003C,0180,0800,0000,124E  ;2548  j/find.mem.lp1  ; Not a memory
U 100C, 0000,003C,6580,0800,0084,7237  ;2549  j/ma780.max,sc_k[.10] ; MA780
U 100D, 0000,003C,0180,0800,0000,124E  ;2550  j/find.mem.lp1  ; Not a memory
U 100E, 0010,0038,0180,09F0,0000,123B  ;2551  rc[0e]_lc,j/found.ms780e ; MS780-E
U 100F, 0010,0038,0180,09F0,0000,123B  ;2552  rc[0e]_lc,j/found.ms780e ; MS780-E
          ;2553 ;
          ;2554 ; Found an MS780-A or -C. Set the starting address
          ;2555 ; to maximum.
          ;2556 ;
U 1231, 0818,0038,5D80,0800,0000,1233  ;2557 MS780.A:d_k[.7],j/ms780.common ; Setup bits<16:14> for -A controller
U 1232, 0818,0038,0580,0800,0000,1233  ;2558 MS780.C:d_k[.1]   ; Setup bits<16:14> for -C controller
          ;2559 MS780.COMMON:
U 1233, 0819,0030,7180,0800,0000,1234  ;2560  d_d.or.k[.fff8]  ; Setup bits<29:17> for either controller
U 1234, 0000,003C,01F8,0803,0080,D235  ;2561  va_va+4,sc_sc+1,q_0 ; Point va to register B, set sc to 14(D)
U 1235, 0D00,003C,0180,0800,0000,1236  ;2562  d_dal.sc  ; Put data in bits<29:14>
          ;2563  cache.p_d[long], ; Set the starting address to maximum
U 1236, 0000,003C,0180,A800,0000,124E  ;2564       j/find.mem.lp1 ; and continue TR scan
          ;2565 ;
          ;2566 ; Found an MA780. Set the starting address to maximum
          ;2567 ;
          ;2568 MA780.MAX:
U 1237, 0818,0038,39F8,0803,0000,1238  ;2569  d_k[.7ff0],q_0,va_va+4 ; Get data for bits<31:20>
U 1238, 0D00,003C,0180,0803,0000,1239  ;2570  d_dal.sc,va_va+4 ; Put in proper position
U 1239, 0000,003C,0180,0803,0000,123A  ;2571  va_va+4   ; Point to Port Invalidate Ctrl Register
          ;2572  cache.p_d[long], ; Set starting address to maximum
U 123A, 0000,003C,0180,A800,0000,124E  ;2573       j/find.mem.lp1
          ;2574 ;
          ;2575 ; Found an MS780E
          ;2576 ;
          ;2577 FOUND.MS780E:
U 123B, 0000,003C,F9F0,2C00,0000,123C  ;2578  q_id[3e]  ; Set up to see how many MS780-E's
          ;2579  alu_q.xor.k[.3], ; Are there Maximum units?
U 123C, 0019,2020,0D80,0800,0010,123D  ;2580   clk.ubcc ; Check condition codes
U 123D, 0000,013C,0180,0800,0000,1098  ;2581  z?   ; Are we at maximum units for system
U 1098, 0000,003C,0180,0800,0000,123E  ;2582 =0 J/ms780e.tst  ; No go find how many units ther are
U 1099, 0818,0038,C180,0800,0000,109A  ;2583  d_k[.ffff]  ; Yes set address to maximum
U 109A, 0000,003D,0180,0800,0000,1256  ;2584 =0 call[set.start.addr] ; .....
U 109B, 0000,003C,0180,0800,0000,124E  ;2585  j/find.mem.lp1  ; Go check to see if we are done
          ;2586 ;
          ;2587 MS780E.TST:
          ;2588  alu_rc[01],  ; Check "Found MS780E Flag"
U 123E, 0010,0038,0180,0908,0D10,123F  ;2589   clk.ubcc ; Check condition codes
U 123F, 0810,0138,0180,0970,0000,109C  ;2590  z?,d_rc[0e]  ; Is this unit the first MS780E ?
          ;2591 =0 j/not.first.ms780e, ; No
U 109C, 0818,0038,C180,0800,0000,10A0  ;2592   d_k[.ffff] ; Set starting address
U 109D, 0001,003C,0180,0988,0000,1240  ;2593  rc[01]_d  ; Yes put base address in rc[01]
U 1240, 0818,0038,7980,0800,0000,1241  ;2594  d_k[.30]  ; Set up SC to point to first unit
U 1241, 0019,0014,F580,0800,0082,1242  ;2595  alu_d+k[.a],sc_alu ; ...
U 1242, 0810,0038,0180,0908,0000,1243  ;2596  d_rc[01]  ; Put base address of unit in D
U 1243, 0000,003C,0180,3400,0000,1244  ;2597  id(sc)_d  ; Put base address in ID pointed to by SC
```

```
U 1244, 0F00,003C,0180,0800,0000,109E  ;2598  d_0   ; Prepare to set starting address to Zero
U 109E, 0000,003D,0180,0800,0000,1256  ;2599 =0 call[set.start.addr] ; Go do it
          ;2600  j/find.mem.lp1.  ; Check to see if we are done
U 109F, 0003,003C,0180,09E8,0000,124E  ;2601   rc[0d]_0 ; ....
          ;2602 =0
          ;2603 NOT.FIRST.MS780E:
U 10A0, 0000,003D,0180,0800,0000,1256  ;2604  call[set.start.addr] ; Go set starting address to maximum
U 10A1, 0000,003C,F9F0,2C00,0000,1245  ;2605  q_id[3e]  ; Get the number of MS780-Es found
U 1245, 0819,2014,0580,0800,0000,1246  ;2606  d_q+k[.1]  ; Increment this counter
U 1246, 0000,003C,F980,3C00,0000,1247  ;2607  id[3e]_d  ; Save the counter
U 1247, 0018,0038,11C0,0800,0000,1248  ;2608  q_k[.4]   ; Prepare to form pointer to the ID registers
          ;2609     ; ...were the I/O base addresses will be stored
U 1248, 081D,2000,7980,0800,0000,1249  ;2610  d_q-d,k[.30]  ; ... adjust pointer
U 1249, 0819,0014,7980,0800,0000,124A  ;2611  d_d+k[.30]  ; Point the SC to the ID register
U 124A, 0000,003C,F580,0800,0000,124B  ;2612  k[.a]   ; ...to receive I/O base address
U 124B, 0019,0014,F580,0800,0082,124C  ;2613  alu_d+k[.a],sc_alu ; ...
U 124C, 0810,0038,0180,0970,0000,124D  ;2614  d_rc[0e]  ; Get base address to d
U 124D, 0000,003C,0180,3400,0000,124E  ;2615  id(sc)_d  ; Move base address to ID pointed to by SC
          ;2616
          ;2617 ;
          ;2618 ; Try next tr
          ;2619 ;
          ;2620 FIND.MEM.LP1:
U 124E, 0818,0014,0580,0A80,0000,124F  ;2621  r[0]_ia+k[.1],d_alu ; Increment TR level
          ;2622  alu_d.and.k[.f],
U 124F, 0019,0034,6180,0800,0010,1250  ;2623      clk.ubcc  ; Check if all done
U 1250, 0000,013C,0180,0800,0000,10A2  ;2624  z?   ; Branch if yes
U 10A2, 0000,003C,0180,0800,0000,1092  ;2625 =0 j/find.mem.loop  ; Try next TR
U 10A3, 0810,0038,0180,0908,0010,1251  ;2626  d_rc[01],clk.ubcc ; Was there any MS780-E'S found ?
U 1251, 0000,013C,0180,0800,0000,10A4  ;2627  z?   ; Check condition codes
U 10A4, 0001,003E,0180,09F0,0000,0002  ;2628 =0 return2,rc[0e]_d ; Yes MS780E was found
U 10A5, 0000,003E,0180,0800,0000,0001  ;2629  return1   ; No MS780E found
          ;2630
```

```
            ;2631  .PAGE
            ;2632  .TOC "   Generate IO Address"
            ;2633 ;++
            ;2634 ; FUNCTIONAL DESCRIPTION:
            ;2635 ;
            ;2636 ; This routine is used to generate an SBI Configuration Register
            ;2637 ; address from a TR level.
            ;2638 ;
            ;2639 ; CALLING CONSTRAINT:
            ;2640 ;
            ;2641 ; =0
            ;2642 ;
            ;2643 ; INPUTS:
            ;2644 ;
            ;2645 ; D - Contains TR level
            ;2646 ;
            ;2647 ; OUTPUTS:
            ;2648 ;
            ;2649 ; D - Contains SBI IO address
            ;2650 ;--
            ;2651 GENERATE.IO:
U 1252. 0000.003C.89F8.0800.0084.7253  ;2652  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 1253. 0D00.003C.8D80.0800.0084.7254  ;2653  d_dal.sc.sc_k[.1f] ; Put TR level in place, setup to
U 1254. 0000.003C.0980.0800.0084.B255  ;2654  sc_sc-k[.2]  ; insert bit 29
U 1255. 0801.001E.0180.0800.0000.0001  ;2655  d_d.ornot.mask,return1 ; and return
            ;2656
```

```
            ;2657  .PAGE
            ;2658  .TOC "   Set Starting Address"
            ;2659 ;**
            ;2660 ; FUNCTIONAL DESCRIPTION:
            ;2661 ;
            ;2662 ; This routine is used to set the starting address of the
            ;2663 ; memory to the specified value.
            ;2664 ;
            ;2665 ; CALLING CONSTRAINT:
            ;2666 ;
            ;2667 ; =0
            ;2668 ;
            ;2669 ; INPUTS:
            ;2670 ;
            ;2671 ; d - Contains address to set memory to (1 Megabyte Increments).
            ;2672 ;     (B Register Image divided by 2**16)
            ;2673 ; rc[0] - Contains Address of Register A
            ;2674 ;
            ;2675 ; OUTPUTS:
            ;2676 ;
            ;2677 ; d - Contains aligned starting address (B Register Image)
            ;2678 ;
            ;2679 ; SIDE EFFECTS:
            ;2680 ;
            ;2681 ; d,q,va, and sc are destroyed
            ;2682 ;--
            ;2683 SET.START.ADDR:
U 1256, 0010,0038,6580,0970,0284,7257  ;2684  va_rc[0e],sc_k[.10] ; Load address of memory Register A
U 1257, 0000,003C,01F8,0803,0000,1258  ;2685  va_va+4,q_0  ; Set address to Register B
            ;2686  d_dal.sc,  ; Put d<15:0> into d<31:16>
U 1258, 0D00,003C,0980,0800,0084,B259  ;2687      sc_sc-k[.2]  ; Setup to insert bit 14
U 1259, 0801,001C,0180,0800,0000,125A  ;2688  d_d.ornot.mask  ; Insert Write Enable bit
U 125A, 0000,003E,0180,A800,0000,0001  ;2689  cache.p_d[long],return1 ; Set the starting address and return
            ;2690
            ;2691
```

```
;2692  .PAGE
;2693  .TOC "   CONFIGURE MS780-E/H"
;2694 ;*************************************************************
;2695 ;++
;2696 ;
;2697 ; Functional Description:
;2698 ; ----------------------
;2699 ;
;2700 ;      This  subroutine  will  be  used  by  tests  that   do   not
;2701 ; specifically  check  the  memory, but  make use of it to execute.
;2702 ; Its purpose is to find a MS780-E and  configure  it  properly  so
;2703 ; that  if  a  Read/Write operation will take place no false errors
;2704 ; will be logged due to misconfigured memory.
;2705 ;
;2706 ; Calling Constraint: =00
;2707 ; ------------------
;2708 ;
;2709 ;
;2710 ; Inputs:
;2711 ; ------
;2712 ;
;2713 ; RC[0E] - I/O BASE OF MS780-E/H
;2714 ;
;2715 ;
;2716 ; Outputs:
;2717 ; -------
;2718 ;
;2719 ; RETURN1 - IF MEMORY IS PROPERLY CONFIGURED
;2720 ; ERORR1  - IF COULD NOT CONFIGURE A MS780-E/H
;2721 ;      OR IF NO MS780-E/Hs WERE FOUND
;2722 ;      ON THE SBI
;2723 ;
;2724 ; Side Effects:
;2725 ; ------------
;2726 ;
;2727 ; SEE INDIVIDUAL SUBROUTINES THAT ARE CALLED.
;2728 ;
;2729 ;
;2730 ;
;2731 ;--
;2732 ;*************************************************************
;2733 CONFIG.MS780E:
U 125B, 0818.0038.0D80.0800.0000.125C  ;2734  D_K[.3]    ; Set up flag for the Fail Chain
U 125C, 0001.003C.0180.09E8.0000.1004  ;2735  RC[0D]_D   ; ...
U 1004, 0000.003D.0180.0800.0000.1090  ;2736 =00 CALL[FIND.MS780E]  ; Any MS780-E/H's on the SBI?
U 1005, 0000.003D.0980.0800.0084.705C  ;2737  ERROR1[.2]   ; No MS780-E/H's found on the SBI
U 1006, 0000.003C.0180.0800.0000.1010  ;2738  NOP     ; OK. Found at least one MS780-E/H
;2739 =
;2740 CONFIG.RETRY:     ; Entry point for the case in which the
;2741       ; ...first MS780-E/H could not be
;2742       ; ...properly configured
U 1010, 0000.003D.0180.0800.0000.1034  ;2743 =00 CALL[SELECT.LOWER]  ; Try to select the lower controller
U 1011, 0000.003C.0180.0800.0000.1018  ;2744  J/CNFG.LMIS   ; Lower controller misconfigured
U 1012, 0000.003E.0180.0800.0000.0001  ;2745  RETURN1     ; OK. Lower Controller is configured
;2746 =
```

```
            ;2747 =00
            ;2748 CNFG.LMIS:
U 1018, 0000,003D,0180,0800,0000,1038  ;2749  CALL[SELECT.UPPER]  ; Select the upper controller
U 1019, 0000,003C,0180,0800,0000,1030  ;2750  J/CNFG.UMIS   ; Upper Controller Misconfigured
U 101A, 0000,003E,0180,0800,0000,0001  ;2751  RETURN1    ; OK. Upper Controller is configured
            ;2752 =
            ;2753 =00
            ;2754 CNFG.UMIS:
U 1030, 0000,003D,0180,0800,0000,125E  ;2755  CALL[ENABLE.NEXT.MS780E] ; Could not configure the first
            ;2756       ; ...MS780-E/H found so go and see
            ;2757       ; ...if there are any more
U 1031, 0000,003C,0180,0800,0000,125D  ;2758  J/CONFIG.RETRY   ; Found another MS780-E/H. Go and
            ;2759       ; ...try to configure it
U 1032, 0818,0038,0D80,0800,0000,1033  ;2760  D_K[.3]    ; Set Flag for Fail Chain
U 1033, 0001,003C,0180,09E8,0000,125D  ;2761  RC[0D]_D    ; ...
U 125D, 0000,003D,0980,0800,0084,705C  ;2762  ERROR1[.2]   ; Could not configure any MS780-E/H
            ;2763       ; ...abort the test
            ;2764 =
            ;2765
```

```
            ;2766  .PAGE
            ;2767  .TOC "   ENABLE NEXT MS780-E"
            ;2768 ;**
            ;2769 ; FUNCTIONAL DESCRIPTION:
            ;2770 ;
            ;2771 ;          This diagnostic will  be able to handle up  to  four  MS780-E units.
            ;2772 ;    They will be tested  separately, according to the TR level at which
            ;2773 ;    they are located,  starting with the one  at the lowest level first.
            ;2774 ;          When  a test has finished  with the first unit, it will have to call
            ;2775 ;    this specific  routine  to see  if any other MS780-E unit is present
            ;2776 ;    on the SBI. If more units are  present, the first one  will be  dis-
            ;2777 ;    abled  by  setting  its  starting  address to maximum, the next unit
            ;2778 ;    will be  enabled  by  setting its starting address to 0 and the test
            ;2779 ;    will be restarted. Subroutine  FIND.MS780E  will  look  on  the  SBI
            ;2780 ;    for  MS780-E/H  subsystems, and will leave their I/O base address in
            ;2781 ;    ID registers 3A through 3D and  a count of the Total number of units
            ;2782 ;    found - 1 in register 3E. In the subroutine the SC register is used
            ;2783 ;    as  a  pointer  to  ID  registers 3A through 3D.
            ;2784 ;
            ;2785 ; CALLING CONSTRAINT:
            ;2786 ;
            ;2787 ;  =00
            ;2788 ;
            ;2789 ;--
            ;2790 ENABLE.NEXT.MS780E:
U 125E, 0000,003C,F9F0,2C00,0000,125F  ;2791  Q_ID[3E]  ; Get total number of MS780E to Q
            ;2792  ALU_Q,   ; Check to see if it is zero
U 125F, 0001,203C,0180,0800,0010,1260  ;2793      CLK.UBCC  ; Check Condition Codes
U 1260, 0000,013C,0180,0800,0000,10A6  ;2794  Z?   ; ...for zero
U 10A6, 0000,003C,0180,0800,0000,1261  ;2795 =0 J/ENABLE.NEXT  ; Not zero go enable next unit
U 10A7, 0000,003E,0180,0800,0000,0002  ;2796  RETURN2   ; Zero No more units to test
            ;2797 ENABLE.NEXT:
U 1261, 0818,0038,C180,0800,0000,10A8  ;2798  D_K[.FFFF]  ; Load D with Maximum Starting address
U 10A8, 0000,003D,0180,0800,0000,1256  ;2799 =0 CALL[SET.START.ADDR] ; Go set starting address of unit to maxium
U 10A9, 0818,0038,7980,0800,0000,1262  ;2800  D_K[.30]  ; Prepare to point to the ID register holding
            ;2801      ; ...the I/O Base address of the next MS780-E
U 1262, 0819,0014,1180,0800,0000,1263  ;2802  D_D+K[.4]  ; ...
U 1263, 0000,003C,F580,0800,0000,1264  ;2803  K[.A]   ; ...
U 1264, 0819,0014,F580,0800,0000,1265  ;2804  D_D+K[.A]  ; ...
U 1265, 0000,003C,F9F0,2C00,0000,1266  ;2805  Q_ID[3E]  ; Get MS780-E Counter
            ;2806  D_D-Q,   ; D now holds the pointer to the ID register
U 1266, 081D,0000,0180,0800,0082,1267  ;2807   SC_ALU  ; ...
U 1267, 0000,003C,01F0,2400,0000,1268  ;2808  Q_ID(SC)  ; Q gets adrress of next MS780E
U 1268, 0001,203C,0180,09F0,0000,1269  ;2809  RC[0E]_Q  ; Save it also in RC[0E]
U 1269, 0F03,003C,0180,09E8,0000,10AA  ;2810  RC[0D]_0,D_0  ; Set starting address to zero
U 10AA, 0000,003D,0180,0800,0000,1256  ;2811 =0 CALL[SET.START.ADDR] ; ....
U 10AB, 0F00,003C,F9F0,2C00,0000,126A  ;2812  Q_ID[3E],D_0  ; Prepare to subtract one from total
U 126A, 081D,2008,0180,0800,0000,126B  ;2813  D_Q-D-1   ; Subtract 1 from total
U 126B, 0000,003C,F980,3C00,0000,10AC  ;2814  ID[3E]_D  ; Adjust the pointer
U 10AC, 0000,003D,0180,0800,0000,11BB  ;2815 =0 CALL[RESET.SUBTST] ; Reset the subtest number
U 10AD, 0000,003E,0180,0800,0000,0001  ;2816  RETURN1   ; Tell caller another unit was found
            ;2817
            ;2818
```

```
            ;2819  .PAGE
            ;2820  .TOC "   Select Controller"
            ;2821 ;++
            ;2822 ; FUNCTIONAL DESCRIPTION:
            ;2823 ;
            ;2824 ; This routine is used to put the memory system into the
            ;2825 ; specified configuration with respect to controller select.
            ;2826 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2827 ; a RETURN2 is made.
            ;2828 ;
            ;2829 ; CALLING CONSTRAINT:
            ;2830 ;
            ;2831 ; =00
            ;2832 ;
            ;2833 ; INPUTS:
            ;2834 ;
            ;2835 ; d - Contains value to write to bits<2:0> of Register A
            ;2836 ; rc[0e] - Address of Register A
            ;2837 ;
            ;2838 ; SIDE EFFECTS:
            ;2839 ;
            ;2840 ; sc,d,va are destroyed
            ;2841 ;--
            ;2842 SELECT.CNTRLR:
U 126C, 0010.0038.0180.0970.0284.726D  ;2843  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 126D, 0801.001C.0180.0800.0000.126E  ;2844  d_d.ornot.mask  ; Insert the enable bit into D reg
U 126E, 0000.003C.0180.A800.0000.126F  ;2845  cache.p_d[long]  ; Write the configuration Register
U 126F, 0818.0038.5D80.0800.0000.1270  ;2846  d_k[.7]   ; Get Misconfiguration bits
U 1270, 0000.003C.61F8.0800.0084.7271  ;2847  sc_k[.F],q_0  ; ...
U 1271, 0D00.003C.0180.0800.0000.1272  ;2848  d_dal.sc  ; ... D = x38000
U 1272, 0000.003C.01E0.0800.0000.1273  ;2849  q_d   ; Save mask
U 1273, 0000.003C.0180.C800.0000.1274  ;2850  d[long]_cache.p  ; Read CNFG A Reg and
            ;2851  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 1274, 001D.2034.0180.0800.0010.1275  ;2852   clk.ubcc ; ...
U 1275, 0000.013C.0180.0800.0000.10AE  ;2853  z?   ; Branch if no
U 10AE, 0000.003E.0180.0800.0000.0001  ;2854 =0 RETURN1   ; Bad configuration
U 10AF, 0000.003E.0180.0800.0000.0002  ;2855  RETURN2   ; Configuration ok
            ;2856
            ;2857
```

```
            ;2858  .PAGE
            ;2859  .TOC "     SELECT LOWER CONTROLLER"
            ;2860 ;*************************************************************
            ;2861 ;**
            ;2862 ;
            ;2863 ; Functional Description:
            ;2864 ; ----------------------
            ;2865 ;
            ;2866 ; This subroutine can be called to select the lower
            ;2867 ; Controller on a MS780-E/H.
            ;2868 ; If the Lower controller is misconfigured then a
            ;2869 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2870 ;
            ;2871 ; Calling Constraint: =00
            ;2872 ; ------------------
            ;2873 ;
            ;2874 ;
            ;2875 ; Inputs:
            ;2876 ; ------
            ;2877 ;
            ;2878 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2879 ;
            ;2880 ; Outputs:
            ;2881 ; -------
            ;2882 ;
            ;2883 ; RC[0D] will have the address of CNFG Register C
            ;2884 ;        ( 2000x008 )
            ;2885 ;
            ;2886 ; Side Effects:
            ;2887 ; ------------
            ;2888 ;
            ;2889 ; Contents of the following registers will lost:
            ;2890 ;
            ;2891 ; D
            ;2892 ;
            ;2893 ; The Lower controller on the MS780-E/H will be configured
            ;2894 ; when the subroutine is exitted with a RETURN2
            ;2895 ;--
            ;2896 ;*************************************************************
            ;2897 =00
            ;2898 SELECT.LOWER:
            ;2899  D_K[ZERO],   ; Interleave mode value for CNFG Reg A
U 1034, 0818,0039,1980,0800,0000,126C  ;2900   CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1035, 0000,003E,0180,0800,0000,0001  ;2901  RETURN1    ; Lower Controller Misconfigured
U 1036, 0810,0038,0180,0970,0000,1037  ;2902  D_RC[0E]   ; Get MS780-E/H I/O Base address
            ;2903  RC[0D] D+K[.8],   ; Set the address of CNFG Reg D
U 1037, 0019,0016,0180,09E8,0000,0002  ;2904   RETURN2   ; Let caller know that the controller
            ;2905       ; ...was configured properly
```

```
            ;2906  .PAGE
            ;2907  .TOC "   SELECT UPPER CONTROLLER"
            ;2908 ;*************************************************************
            ;2909 ;++
            ;2910 ;
            ;2911 ; Functional Description:
            ;2912 ; ----------------------
            ;2913 ;
            ;2914 ; This subroutine can be called to select the upper
            ;2915 ; Controller on a MS780-E/H.
            ;2916 ; If the Upper controller is misconfigured then a
            ;2917 ; RETURN1 will be made.  Otherwise a RETURN2
            ;2918 ;
            ;2919 ; Calling Constraint: =00
            ;2920 ; ------------------
            ;2921 ;
            ;2922 ;
            ;2923 ; Inputs:
            ;2924 ; ------
            ;2925 ;
            ;2926 ; RC[0E] Must hold the I/O base address of the MS780-E/H
            ;2927 ;
            ;2928 ; Outputs:
            ;2929 ; -------
            ;2930 ;
            ;2931 ; RC[0D] will have the address of CNFG Register D
            ;2932 ;        ( 2000x010 )
            ;2933 ;
            ;2934 ; Side Effects:
            ;2935 ; ------------
            ;2936 ;
            ;2937 ; Contents of the following registers will lost:
            ;2938 ;
            ;2939 ; D
            ;2940 ;
            ;2941 ; The Upper controller on the MS780-E/H will be configured
            ;2942 ; when the subroutine is exitted with a RETURN2
            ;2943 ;--
            ;2944 ;*************************************************************
            ;2945 =00
            ;2946 SELECT.UPPER:
            ;2947   D_K[.2],    ; Interleave mode value for CNFG Reg A
U 1038, 0818,0039,0980,0800,0000,126C  ;2948    CALL[SELECT.CNTRLR] ; Set the Interleave mode bits
U 1039, 0000,003E,0180,0800,0000,0001  ;2949   RETURN1     ; Upper Controller Misconfigured
U 103A, 0810,0038,0180,0970,0000,103B  ;2950   D_RC[0E]    ; Get MS780-E/H I/O Base address
            ;2951   RC[0D]_D+K[.C],    ; Set the address of CNFG Reg D
U 103B, 0019,0016,8580,09E8,0000,0002  ;2952    RETURN2    ; Let caller know that the controller
            ;2953       ; ...was configured properly
            ;2954
            ;2955
```

```
;2956 .PAGE "TEST 217 NEXUS SBI FAULT TEST "
;2957 ;*******************************************************************************
;2958 ;**
;2959 ;
;2960 ;
;2961 ; TEST 1F5 CPU AND NEXUS SBI FAULT TESTS
;2962 ;
;2963 ;
;2964 ; TEST DISCRIPTION
;2965 ;
;2966 ; THIS TEST CHECKS NEXUS'S SBI FAULT DETECTION LOGIC.
;2967 ; THE FOLLOWING FAULTS WILL BE CHECKED FOR:
;2968 ;
;2969 ;  1) P0 PARITY ERROR (WRITE CYCLE)
;2970 ;  2) P1 PARITY ERROR (WRITE CYCLE)
;2971 ;  3) MULTIPLE XMITTER ON BUS (WRITE CYCLE)
;2972 ;  4) WRITE DATA SEQUENCE FAULT
;2973 ;  5) INTERLOCK COMMAND SEQUENCE FAULT(UBA ONLY)
;2974 ;
;2975 ; THE DATA USE UNDER THE FAULT CONDITIONS WILL
;2976 ; NOT BE SHIFT OR FLOATED (AS IN A TEST FOR DATA
;2977 ; INDEPENDENCE).  ONLY THE PARITY WILL BE
;2978 ; REVERSED (AT THE CPU) OR MAINTENANCE I.D
;2979 ; USED.  THE MAINTAINENCE I.D WILL BE PART
;2980 ; OF THE FORCE FAULT CONSTANTS (RA1-4).
;2981 ;
;2982 ; EACH TEST WILL CHECK TO SEE THE SBI FAULT
;2983 ; BIT CLEARED ON THE CPU DEASSERTION OF FAULT.
;2984 ; THE FAULT WILL BE CHECKED IN THE CPU MAINTENANCE
;2985 ; REGISTER AS WELL AS THE NEXUS CONFIGURATION
;2986 ; REGISTER.
;2987 ;
;2988 ;
;2989 ; TEST SEQUENCE
;2990 ; -----------
;2991 ;  SUBTEST 2 P0 FAULT (WRITE CYCLE)
;2992 ;
;2993 ;  SUBTEST 3 P1 FAULT (WRITE CYCLE)
;2994 ;
;2995 ;  SUBTEST 4 WRITE DATA SEQUENCE FAULT (WRITE CYCLE)
;2996 ;
;2997 ;  SUBTEST 5 NORMAL INTERLOCK COMMAND SEQUENCE (UBA ONLY)
;2998 ;  I.E. INTERLOCK READ,INTERLOCK WRITE
;2999 ;         WITH COMPLIMENT DATA
;3000 ;
;3001 ;  SUBTEST 6 INTERLOCK SEQUENCE FAULT( UBA ONLY)
;3002 ;
;3003 ;  SUBTEST 7 INTERLOCK TIMER TEST #1 (UBA ONLY)
;3004 ;  I.E. INTERLOCK READ-INTERLOCK READ AND NO TIMEOUT
;3005 ;
;3006 ;  SUBTEST 8 UNEXPECTED READ DATA TO NEXUS
;3007 ;
;3008 ;  SUBTEST 9 MULTIPLE TRANSMITTER FAULTS ( ALL 32 I.D. USED)
;3009 ;
;3010 ;  SUBTEST A SBI ERROR COMFIRMATION( BYTE WRITES TO I/O SPACE)
```

```
;3011 ;
;3012 ;  SUBTEST B SBI ERROR COMFIRMATION ( EXTENDED WRITES TO I/O SPACE )
;3013 ;
;3014 ;
;3015 ;
;3016 ;
;3017 ; ERROR LOGOUT
;3018 ;
;3019 ; DATA: ITEM 1 (SEE INDIVIDUAL TEST)
;3020 ;  ITEM 2 (SEE INDIVIDUAL TEST)
;3021 ;  DEVICE FLAG
;3022 ;     0 = DW750
;3023 ;     1 = RH750
;3024 ;     2 = CI750
;3025 ;     3 = DR780
;3026 ;  TR LEVEL OF ADAPTER UNDER TEST
;3027 ;  TR LEVEL OF MEMORY BEING USED
;3028 ;
;3029 ; SYNC POINT DESCRIPTION
;3030 ;
;3031 ;
;3032 ;
;3033 ;--
;3034 ;*****************************************************************************
;3035 ;
;3036 ; SCRATCH PAD REQUIREMENT
;3037 ; ----------------------
;3038 ;
;3039 ; RA,B DESCRIPTION
;3040 ; ---- -----------
;3041 ;
;3042 ;   0 CLEAR CPU FAULT BIT MASK,   ^X0008 0000
;3043 ;  CPU FAULT BIT    ^X0008 0000
;3044 ;   1 FORCE P0 SBI ERROR     ^X8000 0000
;3045 ;  PARITY FAULT MASK   ^X8000 0000
;3046 ;   2 FORCE P1 SBI ERROR                  ^X0000 0800
;3047 ;   3 FORCE WRITE SEQ FAULT (INCL MAINT ID) ^X4F80 0000
;3048 ;   4 FORCE MULTIPLE XMITTERS (INCL MAINT ID) ^X1F80 0000
;3049 ;   5
;3050 ;   6 MULTIPLE XMITTER FAULT MASK( REG A)  ^X0800 0000
;3051 ;   7 WRITE SEQUENCE FAULT BIT  ^X4000 0000
;3052 ;   8 CPU FAULT INTERRUPT ENABLE    ^X0004 0000
;3053 ;   9 SBI FAULT MASK    ^X0002 0000
;3054 ;   A CLEAR FORCE ERROR BITS,DISABLE CACHE ^X0001 8000
;3055 ;   B TRANSMITTER DURING FAULT MASK  ^X0400 0000
;3056 ;   C MASK FOR CPU GOT BUS, DEVICE BUSY, TIMEOUT
;3057 ;       ^X0000 1C00
;3058 ;   D DATA1  "0F0F0F0F"
;3059 ;   E DATA2  "F0F0F0F0"
;3060 ;   F U-TRAPS FLAGS
;3061 ;
;3062 ;
;3063 ; RC DESCRIPTION
;3064 ; -- -----------
;3065 ;  3 MEMORY BEING USED CONFIG REG A ADDRESS ^X200YY000
```

```
            ;3066 ;  4 CONSTANT FOR CPU TIMEOUT, GOT BUS, DEVICE BUSY
            ;3067 ;      ^X0000 1400
            ;3068 ;  5 INTERLOCK FAULT MASK (REG A)     ^X1000 0000
            ;3069 ;  6 FAULT BIT MASK    ^XFC00 0000
            ;3070 ;  7 BASE ADDRESS OF UBA SPACE( IN SBI I/O SPACE)
            ;3071 ;  8 UNEXPECTED READ DATA FAULT BIT  ^X2000 0000
            ;3072 ;  9
            ;3073 ;  A TR LEVEL OF MEMROY BEING USED
            ;3074 ;  B TR LEVEL OF ADAPTER UNDER TEST
            ;3075 ;  C DEVICE FLAG (SEE ABOVE)
            ;3076 ;  D SEE INDIVIDUAL TESTS
            ;3077 ;  E SEE INDIVIDUAL TESTS
            ;3078 ;  F NUMBER OF ARGUMENTS TO BE PASSED TO THE CONSOLE
            ;3079 ;
            ;3080 ;*****************************************************************************
            ;3081 =0
U 10B0, 0000,003D,0180,0800,0000,1063  ;3082 M06000: NEWTST
U 10B1, 0000,003C,0180,0800,0000,10B2  ;3083  NOP
U 10B2, 0000,003D,0180,0800,0000,125B  ;3084 =0 CALL[CONFIG.MS780E]  ; Configure a MS780-E/H
U 10B3, 0810,0038,0180,0970,0000,1276  ;3085  D_RC[0E]   ; Get I/O Base address of MS780-E/H
            ;3086       ; ...being used
U 1276, 0001,003C,0180,0998,0000,1277  ;3087  RC[3]_D    ; Save in RC[3] and
U 1277, 0001,003C,0180,09D0,0000,10B4  ;3088  RC[0A]_D   ; ...in RC[0A] for error logout
U 10B4, 0000,003D,0180,0800,0000,11F8  ;3089 =0 CALL[INIT]
            ;3090       ;
U 10B5, 0818,0038,41F8,0800,0000,1278  ;3091  D_K[.80],Q_0
U 1278, 0000,003C,1180,0800,0084,7279  ;3092  SC_K[.4]
U 1279, 0D00,003C,0180,0800,0000,127A  ;3093  D_DAL.SC
U 127A, 0001,003C,0180,0A90,0000,127B  ;3094  R[2]_D     ;RA 2
U 127B, 0D00,003C,0180,0800,0000,127C  ;3095  D_DAL.SC
U 127C, 0000,003C,01E0,0800,0000,127D  ;3096  Q_D
U 127D, 0500,003C,0180,0800,0000,127E  ;3097  D_D.LEFT
U 127E, 0001,003C,0180,0AC8,0000,127F  ;3098  R[9]_D     ;SAVE D REG
U 127F, 081D,0030,0180,0800,0000,1280  ;3099  D_D.OR.Q
U 1280, 0001,003C,01F8,0AD0,0000,1281  ;3100  R[0A]_D,Q_0    ;RA 0A
U 1281, 0800,003C,0180,0A48,0000,1282  ;3101  D_R[9]     ;RESTORE D
U 1282, 0500,003C,0180,0800,0000,1283  ;3102  D_D.LEFT
U 1283, 0001,003C,0180,0AC8,0000,1284  ;3103  R[9]_D     ;RA 9
U 1284, 0500,003C,0180,0800,0000,1285  ;3104  D_D.LEFT
U 1285, 0001,003C,0180,0AC0,0000,1286  ;3105  R[08]_D     ;RA 8
U 1286, 0500,003C,0180,0800,0000,1287  ;3106  D_D.LEFT
U 1287, 0001,003C,0180,0A80,0000,1288  ;3107  R[0]_D     ;RA 0
U 1288, 0000,003C,5D80,0800,0084,7289  ;3108  SC_K[.7]
U 1289, 0D00,003C,0180,0800,0000,128A  ;3109  D_DAL.SC
U 128A, 0001,003C,0180,0AD8,0000,128B  ;3110  R[0B]_D     ;RA B
U 128B, 0500,003C,0180,0800,0000,128C  ;3111  D_D.LEFT
U 128C, 0001,003C,0180,0AB0,0000,128D  ;3112  R[6]_D     ;RA 6
U 128D, 0500,003C,0180,0800,0000,128E  ;3113  D_D.LEFT
U 128E, 0501,003C,0180,09A8,0000,128F  ;3114  RC[05]_D,D_D.LEFT   ;RC 5
U 128F, 0501,003C,0180,09C0,0000,1290  ;3115  RC[8]_D,D_D.LEFT   ;RC 8
U 1290, 0501,003C,0180,0AB8,0000,1291  ;3116  R[7]_D,D_D.LEFT    ;RA 7
U 1291, 0001,003C,0180,0A88,0000,1292  ;3117  R[1]_D     ;RA 1
            ;3118       ;
U 1292, 0818,0038,11F8,0800,0082,1293  ;3119  D_K[.4],Q_0,SC_ALU   ;RA 3
U 1293, 0D00,003C,0180,0800,0000,1294  ;3120  D_DAL.SC
```

```
U 1294. 0819.0030.6180.0800.0000.1295  ;3121  D_D.OR.K[.F]
U 1295. 0D00.003C.2180.0800.0084.7296  ;3122  D_DAL.SC,SC_K[.14]
U 1296. 0819.0030.0180.0800.0000.1297  ;3123  D_D.OR.K[.8]
U 1297. 0D00.003C.0180.0800.0000.1298  ;3124  D_DAL.SC
U 1298. 0001.003C.0180.0A98.0000.1299  ;3125  R[3]_D
          ;3126  ;
U 1299. 0000.003C.5180.0800.0084.729A  ;3127  SC_K[.1E]   ;RA 4
U 129A. 0801.0034.0180.0800.0000.129B  ;3128  D_D.AND.MASK
U 129B. 0000.003C.0980.0800.0084.B29C  ;3129  SC_SC-K[.2]
U 129C. 0801.001C.0180.0800.0000.129D  ;3130  D_D.ORNOT.MASK
U 129D. 0001.003C.0180.0AA0.0000.129E  ;3131  R[4]_D
          ;3132  ;
U 129E. 0818.0038.F1F8.0800.0000.129F  ;3133  D_K[.FFFC],Q_0   ;RC 6
U 129F. 0000.003C.7D80.0800.0084.72A0  ;3134  SC_K[.18]
U 12A0. 0D00.003C.0180.0800.0000.12A1  ;3135  D_DAL.SC
U 12A1. 0001.003C.0180.09B0.0000.12A2  ;3136  RC[6]_D
          ;3137  ;
U 12A2. 0F00.003C.11F8.0800.0084.72A3  ;3138  D_0,SC_K[.4],Q_0  ;RA C
U 12A3. 0819.0030.0580.0800.0000.12A4  ;3139  D_D.OR.K[.1]
U 12A4. 0D00.003C.0180.0800.0084.72A5  ;3140  D_DAL.SC,SC_K[.8]
U 12A5. 0819.0030.8580.0800.0000.12A6  ;3141  D_D.OR.K[.C]
U 12A6. 0D00.003C.0180.0800.0000.12A7  ;3142  D_DAL.SC
U 12A7. 0001.003C.0180.0AE0.0000.12A8  ;3143  R[0C]_D
          ;3144  ;
U 12A8. 0818.0038.CDF8.0800.0000.12A9  ;3145  D_K[.F0],Q_0   ;RA D,E
U 12A9. 0D00.003C.0180.0800.0000.12AB  ;3146  D_DAL.SC
U 12AB. 0819.0030.CD80.0800.0000.12AC  ;3147  D_D.OR.K[.F0]
U 12AC. 0D00.003C.0180.0800.0000.12AD  ;3148  D_DAL.SC
U 12AD. 0819.0030.CD80.0800.0000.12AE  ;3149  D_D.OR.K[.F0]
U 12AE. 0D00.003C.0180.0800.0000.12AF  ;3150  D_DAL.SC
U 12AF. 0819.0030.CD80.0800.0000.12B0  ;3151  D_D.OR.K[.F0]
U 12B0. 0001.003C.0180.0AE8.0000.12B1  ;3152  R[0D]_D    ;RA D
U 12B1. 0801.0028.0180.0AF0.0000.12B2  ;3153  R[0E]_ALU,D_NOT.D
U 12B2. 0F00.003C.8580.0800.0084.72B3  ;3154 ADP020: D_0,SC_K[.C]
U 12B3. 0801.001C.0180.0800.0000.12B4  ;3155  D_D.ORNOT.MASK
U 12B4. 0000.003C.0980.0800.0084.B2B5  ;3156  SC_SC-K[.2]
U 12B5. 0801.001C.0180.0800.0000.12B6  ;3157  D_D.ORNOT.MASK
U 12B6. 0001.003C.0180.09A0.0000.12B7  ;3158  RC[4]_D
U 12B7. 0018.0038.0580.09D8.0000.12B8  ;3159  RC[0B]_K[.1]   ;SET 1ST TR LEVEL TO BE CHECKED
U 12B8. 0F00.003C.8980.0800.0084.72B9  ;3160  D_0,SC_K[.D]
U 12B9. 0801.001C.6580.0800.0000.12BA  ;3161  D_D.ORNOT.MASK,K[.10]
U 12BA. 0000.003C.6580.0800.0084.92BB  ;3162  SC_SC+K[.10]
U 12BB. 0801.001C.0180.0980.0000.12BC  ;3163  RC[0]_ALU,D_D.ORNOT.MASK
U 12BC. 0819.0014.1180.0988.0000.12BD  ;3164  RC[1]_ALU,D_D+K[.4]
U 12BD. 0819.0014.1180.0990.0000.10B6  ;3165  RC[2]_ALU,D_D+K[.4]
          ;3166  ;
          ;3167  ;=============================================================================
          ;3168  ;*
          ;3169  ;FIND THE ADAPTER CODE, IF REGISTER RESPONDS
          ;3170  ;
          ;3171  ; ERROR LOGOUT
          ;3172  ;
          ;3173  ; DATA: TR LEVEL BEING CHECKED
          ;3174  ;  IMAGE OF CONFIGURATION REGISTER READ
          ;3175  ;  DEVICE FLAG,0=UBA,1=MBA,2=YY780
```

```
          :3176  ;  TR LEVEL OF ADAPTER UNDER TEST
          :3177  ;  TR LEVEL OF MEMORY BEING USED
          :3178  ;-
          :3179  ;=============================================================================
          :3180  ;
          :3181 =0
U 10B6. 0000.003D.0180.0800.0000.11F8  ;3182 CLN000: CALL[INIT]
U 10B7. 0003.003C.0180.09F8.0000.10B8  ;3183  RC[0F]_0    ; FORCE SUBTEST COUNTER TO ZERO
U 10B8. 0010.0039.D580.0900.0284.71DA  ;3184 =0 VA_RC[0],SETRCF[.6]  ;SET ADDRESS,SET # ARG TO
U 10B9. 0F00.003C.8980.0800.0084.72BE  ;3185  D_0,SC_K[.D]    ;CATCH A TIME OUT
U 12BE. 0801.001C.0180.0AF8.0000.12BF  ;3186  R[0F]_ALU,D_D.ORNOT.MASK
U 12BF. 0000.003C.0180.C800.0000.12C0  ;3187  MCT/READ.P    ;READ CONFIGURATION REGISTER
U 12C0. 0001.003C.0180.09E8.0000.12C1  ;3188  RC[0D]_D    ;SAVE THE DATA READ
U 12C1. 0000.003C.0180.0A78.0010.12C2  ;3189  ALU_R[0F],CLK.UBCC  ;DID IT TIMEOUT
U 12C2. 0000.013C.0180.0800.0000.10BA  ;3190  Z?
U 10BA. 0000.003C.0180.0800.0000.103C  ;3191 =0 J/CLN010    ;NO TIMEOUT
U 10BB. 0000.003C.0180.0800.0000.139F  ;3192  J/CLN900    ;TIMED OUT , GO CHECK FOR END OF LOOOP
          :3193 =00
U 103C. 0000.003D.0180.0800.0000.11DF  ;3194 CLN010: CALL[NOINTR]
U 103D. 0000.003D.0180.0800.0000.11B7  ;3195  ERROR1     ;UNEXPECTED INTERRUPT !!!!
U 103E. 0810.0038.0180.0958.0000.103F  ;3196  D_RC[0B]    ;GET TR LEVEL UNDER TEST
U 103F. 0001.003C.0180.09F0.0000.12C3  ;3197  RC[0E]_D    ;SET IT TO CONSOLE
U 12C3. 0810.0038.0180.0968.0000.12C4  ;3198  D_RC[0D]    ;GET CONFIG REG A DATA
U 12C4. 0200.003C.0180.0800.0000.12C5  ;3199  D_D.RIGHT2    ; SETUP TO TEST BITS<7:3>
U 12C5. 0000.003C.01E0.0800.0000.12C6  ;3200  Q_D
U 12C6. 0200.003C.0180.0800.0000.12C7  ;3201  D_D.RIGHT2    ; ...
U 12C7. 0C00.193C.0180.0800.0000.1017  ;3202  D3?,D_Q     ; BIT 7 SET?
U 1017. 0000.003C.0180.0800.0000.12C8  ;3203 =0111 J/CLN020    ; BRANCH IF NO
U 101F. 0000.003C.0180.0800.0000.139F  ;3204  J/CLN900    ; UNKNOWN DEVICE
U 12C8. 0600.003C.0180.0800.0000.12C9  ;3205 CLN020: D_D.RIGHT    ; GET BITS<6:3> INTO <3:0> POSITION
U 12C9. 0000.193C.0180.0800.0000.1020  ;3206  D3-0?     ; BRANCH ON BITS<6:3>
U 1020. 0000.003C.0180.0800.0000.139F  ;3207 =0000 J/CLN900    ; UNKNOWN DEVICE
U 1021. 0000.003C.0180.0800.0000.139F  ;3208  J/CLN900    ; MS780
U 1022. 0000.003C.0180.0800.0000.139F  ;3209  J/CLN900    ; MS780
U 1023. 0000.003C.0180.0800.0000.139F  ;3210  J/CLN900    ; MS780
U 1024. 0018.0038.0580.09E0.0000.10BC  ;3211  RC[0C]_K[.1],J/CLN200   ; RH780
U 1025. 0018.0038.1980.09E0.0000.12CA  ;3212  RC[0C]_K[ZERO],J/CLN030   ; DW780
U 1026. 0018.0038.0D80.09E0.0000.10BC  ;3213  RC[0C]_K[.3],J/CLN200   ; DR780
U 1027. 0018.0038.0980.09E0.0000.10BC  ;3214  RC[0C]_K[.2],J/CLN200   ; CI780
U 1028. 0000.003C.0180.0800.0000.139F  ;3215  J/CLN900    ; MA780
U 1029. 0000.003C.0180.0800.0000.139F  ;3216  J/CLN900    ; UNKNOWN
U 102A. 0000.003C.0180.0800.0000.139F  ;3217  J/CLN900    ; UNKNOWN
U 102B. 0000.003C.0180.0800.0000.139F  ;3218  J/CLN900    ; UNKNOWN
U 102C. 0000.003C.0180.0800.0000.139F  ;3219  J/CLN900    ; MS780
U 102D. 0000.003C.0180.0800.0000.139F  ;3220  J/CLN900    ; MS780
U 102E. 0000.003C.0180.0800.0000.139F  ;3221  J/CLN900    ; MS780
U 102F. 0000.003C.0180.0800.0000.139F  ;3222  J/CLN900    ; MS780
U 12CA. 0810.0038.0180.0968.0000.12CB  ;3223 CLN030: D_RC[0D]    ;GET CONFIG REG A IMAGE THAT WAS READ
U 12CB. 0819.0034.0D80.0800.0000.12CC  ;3224  D_D[AND]K[.3]    ;FIND UBA BASE ADDRESS FOR SBI I/O SPACE
U 12CC. 0819.0030.11F8.0800.0000.12CD  ;3225  D_D[OR]K[.4],Q_0
U 12CD. 0000.003C.2180.0800.0084.72CE  ;3226  SC_K[.14]
U 12CE. 0000.003C.0980.0800.0084.B2CF  ;3227  SC_SC-K[.2]
U 12CF. 0D00.003C.D980.0800.0000.12D0  ;3228  D_DAL.SC.K[.9]
U 12D0. 0000.003C.D980.0800.0084.92D1  ;3229  SC_SC+K[.9]
U 12D1. 0000.003C.0980.0800.0084.92D2  ;3230  SC_SC+K[.2]
```

```
U 12D2, 0801,001C,0180,09B8,0000,10BC  ;3231  RC[7]_ALU,D_D.ORNOT.MASK,J/CLN200
          ;3232 ;
          ;3233 ;*****************************************************************************
          ;3234 ;*
          ;3235 ;** SUBTEST 2 **
          ;3236 ;
          ;3237 ;FORCE P0 FAULT
          ;3238 ;
          ;3239 ;SEQUENCE OF STEPS
          ;3240 ; 1)CACHE OFF,FORCE P0 FAULT
          ;3241 ; 2)INITIATE WRITE OPERATION
          ;3242 ; 3)CLEAR FORCE P0 FAULT
          ;3243 ; 4)SEE PARITY BIT SET IN NEXUS REG
          ;3244 ; 5)DEASSERT FAULT BIT
          ;3245 ; 6)CHECK THAT BIT CLEARED IN NEXUS
          ;3246 ;
          ;3247 ;ERROR LOGOUT
          ;3248 ;
          ;3249 ; IF NONE: EITHER THE CPU FALUT OR SBI FAULT BIT IN
          ;3250 ;       ERROR(SEE LISTING)
          ;3251 ;
          ;3252 ;     DATA: EXPECTED SBI-FAULT-STATUS BITS (BITS 31-26)
          ;3253 ;       MASKED SBI-FAULT-STATUS BITS READ(BITS 31-26)
          ;3254 ;       EITHER FROM CPU OR NEXUS(SEE LISTING)
          ;3255 ;    DEVICE FLAG
          ;3256 ;    TR LEVEL OF ADAPTER UNDER TEST
          ;3257 ;    TR LEVEL OF MEMORY BEING USED
          ;3258 ;    UNDEFINED
          ;3259 ;-
          ;3260 ;*****************************************************************************
          ;3261 ;
          ;3262 =0
U 10BC, 0000,003D,0180,0800,0000,11D8  ;3263 CLN200: SUBTEST    ; FORCE SUBTEST COUNTER TO 1
U 10BD, 0000,003C,0180,0800,0000,10BE  ;3264  NOP
U 10BE, 0000,003D,0180,0800,0000,11D8  ;3265 =0 SUBTEST    ; SET SUBTEST COUNTER TO 2
U 10BF, 0000,003C,0180,0800,0000,10C0  ;3266  NOP
U 10C0, 0000,003D,0180,0800,0000,11B6  ;3267 =0 ERLOOP
U 10C1, 0000,003C,0180,0800,0000,10C2  ;3268  NOP
U 10C2, 0000,003D,0180,0800,0000,11F8  ;3269 =0 CALL[INIT]
U 10C3, 0800,003C,0180,0A50,0000,10C4  ;3270  D_R[0A]    ;FORCE THE CACHE OFF,FORCE P0
U 10C4, 080D,0031,D580,0A08,0084,71DA  ;3271 =0 D_ALU,ALU_D[OR]R[1],SETRCF[.6]
U 10C5, 0000,003C,7580,3C00,0000,12D3  ;3272  ID[MAINT]_D   ;LOAD SBI MAINTENANCE REG
U 12D3, 0010,0038,0180,0900,0200,12D4  ;3273  VA_RC[0]
U 12D4, 0F00,003C,B980,0800,0084,72D5  ;3274  D_0,SC_K[.D]   ;CATCH TIMEOUT
U 12D5, 0801,001C,0180,0AF8,0000,12D6  ;3275  R[0F]_ALU,D_D.ORNOT.MASK
U 12D6, 0000,003C,0180,A800,0000,12D7  ;3276  MCT/WRITE.P
U 12D7, 0800,003C,0180,0A50,0000,12D8  ;3277  D_R[0A]    ;CLEAR THE FORCE FAULT BIT
U 12D8, 0000,003C,7580,3C00,0000,10C6  ;3278  ID[MAINT]_D
          ;3279 =0
U 10C6, 0000,003D,D580,0800,0084,71DA  ;3280 M06050: SETRCF[.6]
U 10C7, 0010,0038,0180,0900,0200,12D9  ;3281  VA_RC[0]   ;DID NEXUS SEE FAULT
U 12D9, 0000,003C,0180,C800,0000,12DA  ;3282  MCT/READ.P   ;READ CONFIG A REG
U 12DA, 0811,0034,0180,0930,0000,12DB  ;3283  D_ALU,ALU_D[AND]RC[06]  ;MASK FAULT BITS
U 12DB, 0001,003C,0180,09E8,0000,12DC  ;3284  RC[0D]_D   ;SAVE MASKED IMAGE
U 12DC, 0000,003C,01C0,0A08,0000,12DD  ;3285  Q_R[1]
```

```
ZZ-ESKAR-V22  TEST 217 NEXUS SBI FAULT TEST                                                    SEQ 2223
ESKAR58.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40                          Page   8

U 12DD. 0011.2020.0180.0968.0010.12DE  ;3286  ALU_Q[XOR]RC[0D].CLK.UBCC ;CHECK THE PARITY FAULT BIT
U 12DE. 0001.213C.0180.09F0.0000.10C8  ;3287  Z?,RC[0E]_Q
U 10C8. 0000.003D.0180.0800.0000.11B7  ;3288 =0 ERROR1     ;WRONG FAULT BITS SET
U 10C9. 0800.003C.0180.0A00.0000.10CA  ;3289  D_R[0]
U 10CA. 0000.003D.D580.0800.0084.71DA  ;3290 =0 SETRCF[.6]
U 10CB. 0000.003C.6D80.3C00.0000.12DF  ;3291  ID[FAULT]_D
U 12DF. 0010.0038.0180.0900.0200.12E0  ;3292  VA_RC[0]    ; CHECK THAT BIT CLEARED IN NEXUS
U 12E0. 0000.003C.0180.C800.0000.12E1  ;3293  MCT/READ.P    ;READ TO D REG
U 12E1. 0811.0034.0180.0930.0000.12E2  ;3294  D_ALU,ALU_D[AND]RC[6]   ;MASK FAULT BITS
U 12E2. 0001.003C.0180.09E8.0000.12E3  ;3295  RC[0D]_D    ;SAVE MASKED BITS
U 12E3. 0810.0038.0180.0968.0010.12E4  ;3296  D_RC[0D],CLK.UBCC
U 12E4. 0000.013C.0180.0800.0000.10CC  ;3297  Z?
U 10CC. 0000.003D.0180.0800.0000.11B7  ;3298 =0 ERROR1     ;SBI FAULT STATUS BITS SET
U 10CD. 0000.003C.0180.0800.0000.10CE  ;3299  NOP
          ;3300 ;
          ;3301 ;*************************************************************************
          ;3302 ;*
          ;3303 ;** SUBTEST 3 **
          ;3304 ;
          ;3305 ;CHECK P1 FAULT LOGIC
          ;3306 ;
          ;3307 ;SEQUENCE OF STEPS
          ;3308 ; 1) LOAD MAINTENANCE REGISTER WITH FORCE P1 ERROR
          ;3309 ; 2) WRITE TO NEXUS
          ;3310 ; 3) CLEAR FORCE P1 IN MAINTENANCE
          ;3311 ; 4) READ NEXUS CONFIG REG A SEE PARITY FAULT
          ;3312 ; 5) CLEAR CPU FAULT BIT
          ;3313 ; 6) SEE NEXUS CONFIG REG A PARITY FAULT CLEARED
          ;3314 ;
          ;3315 ;ERROR LOGOUT
          ;3316 ;
          ;3317 ; IF NONE: EITHER THE CPU FALUT OR SBI FAULT BIT IN
          ;3318 ;       ERROR(SEE LISTING)
          ;3319 ;
          ;3320 ;     DATA: EXPECTED SBI-FAULT-STATUS BITS (BITS 31-26)
          ;3321 ;       MASKED SBI-FAULT-STATUS BITS READ(BITS 31-26)
          ;3322 ;       EITHER FROM CPU OR NEXUS(SEE LISTING)
          ;3323 ;    DEVICE FLAG (SEE TEST HEADER)
          ;3324 ;    TR LEVEL OF ADAPTER UNDER TEST
          ;3325 ;    TR LEVEL OF MEMORY BEING USED
          ;3326 ;    UNDEFINED
          ;3327 ;-
          ;3328 ;*************************************************************************
          ;3329 ;
U 10CE. 0000.003D.0180.0800.0000.11D8  ;3330 =0 SUBTEST
U 10CF. 0000.003C.0180.0800.0000.10D0  ;3331  NOP
U 10D0. 0000.003D.0180.0800.0000.11B6  ;3332 =0 ERLOOP
U 10D1. 0000.003C.0180.0800.0000.10D2  ;3333  NOP
U 10D2. 0000.003D.0180.0800.0000.11F8  ;3334 =0 CALL[INIT]
U 10D3. 0800.003C.0180.0A10.0000.12E5  ;3335  D_R[2]     ;GENERATE CACHE OFF AND FORCE
U 12E5. 080D.0030.0180.0A50.0000.12E6  ;3336  D_ALU,ALU_D[OR]R[0A]   ; P1 FAULT
U 12E6. 0000.003C.7580.3C00.0000.10D4  ;3337  ID[MAINT]_D
U 10D4. 0000.003D.D580.0800.0084.71DA  ;3338 =0 SETRCF[.6]
U 10D5. 0F10.0038.8980.0900.0284.72E7  ;3339  VA_RC[0],D_0,SC_K[.D]   ;CATCH THE TIMEOUT
U 12E7. 0801.001C.0180.0AF8.0000.12E8  ;3340  R[0F]_ALU,D_D.ORNOT.MASK
```

```
U 12E8. 0000.003C.0180.A800.0000.12E9  ;3341  MCT/WRITE.P   ;DO THE WRITE OP
U 12E9. 0800.003C.0180.0A50.0000.12EA  ;3342  D_R[0A]    ;CLEAR FORCE FAULT
U 12EA. 0000.003C.7580.3C00.0000.10D6  ;3343  ID[MAINT]_D
        ;3344 =0
U 10D6. 0010.0039.D580.0900.0284.71DA  ;3345 M06120: SETRCF[.6],VA_RC[0]  ;DID NEXUS SEE PARITY FAULT ?
U 10D7. 0000.003C.0180.C800.0000.12EB  ;3346  MCT/READ.P   ;READ AND
U 12EB. 0811.0034.0180.0930.0000.12EC  ;3347  D_ALU,ALU_D[AND]RC[06]  ;MASK FAULT BITS
U 12EC. 0001.003C.0180.09E8.0000.12ED  ;3348  RC[0D]_D   ;SAVE IMAGE
U 12ED. 0000.003C.01C0.0A08.0000.12EE  ;3349  Q_R[1]
U 12EE. 0011.2020.0180.0968.0010.12EF  ;3350  ALU_Q[XOR]RC[0D],CLK.UBCC ;TEST NEXUS SBI FAULT STATUS
U 12EF. 0001.213C.0180.09F0.0000.10D8  ;3351  Z?,RC[0E]_Q   ;SET EXPECTED DATA
U 10D8. 0000.003D.0180.0800.0000.11B7  ;3352 =0 ERROR1    ;WRONG BITS SET
U 10D9. 0800.003C.0180.0A00.0000.10DA  ;3353 M06160: D_R[0]
U 10DA. 0000.003D.D580.0800.0084.71DA  ;3354 =0 SETRCF[.6]
U 10DB. 0000.003C.6D80.3C00.0000.12F0  ;3355  ID[FAULT]_D
U 12F0. 0010.0038.0180.0900.0200.10DC  ;3356  VA_RC[0]
U 10DC. 0003.003D.D580.09F0.0084.71DA  ;3357 =0 SETRCF[.6],RC[0E]_0
U 10DD. 0000.003C.0180.C800.0000.12F1  ;3358  MCT/READ.P   ;READ CONFIG REG A TO D REG
U 12F1. 0811.0034.0180.0930.0000.12F2  ;3359  D_ALU,ALU_D[AND]RC[6]  ;MASK FAULT BITS
U 12F2. 0001.003C.0180.09E8.0010.12F3  ;3360  RC[0D]_D,CLK.UBCC  ;SAVE MASKED BITS
U 12F3. 0000.013C.0180.0800.0000.10DE  ;3361  Z?
U 10DE. 0000.003D.0180.0800.0000.11B7  ;3362 =0 ERROR1    ;SBI FAULT STATUS BITS SET
U 10DF. 0000.003C.0180.0800.0000.10E0  ;3363  NOP
        ;3364 ;
        ;3365 ;*************************************************************************
        ;3366 ;*
        ;3367 ;** SUBTEST 4 **
        ;3368 ;
        ;3369 ; CHECKS THE WRITE SEQUENCE(WS)
        ;3370 ; FAULT.  FORCES THE FAULT FROM THE
        ;3371 ; CPU, AND EXAMINES NEXUS
        ;3372 ; CONFIGURATION REGISTER A TO SEE THE
        ;3373 ; FAULT DETECTED.  ALSO CHECKS CPU FAULT/STATUS
        ;3374 ; REGISTER TO SEE THE SBI FAULT WAS LOGGED.
        ;3375 ;
        ;3376 ;
        ;3377 ;SEQUENCE OF OPERATION
        ;3378 ;=========================================================================
        ;3379 ; CLEAR AND CHECK CPU FAULT BIT
        ;3380 ;  (DONE DURING PREVIOUS TEST)
        ;3381 ; SET FORCE WRITE SEQUENCE FAULT BIT
        ;3382 ;  IN MAINTENANCE REGISTER
        ;3383 ; START A WRITE OPERATION TO NEXUS( CATCH THE TIMEOUT)
        ;3384 ; DEASSERT FORCE FAULT
        ;3385 ; CHECK CPU FAULT REGISTER
        ;3386 ;  A) SEE CPU FAULT SET
        ;3387 ;  B) SEE SBI FAULT SET
        ;3388 ; READ NEXUS CONFIGURATION REG A
        ;3389 ;  SEE WRITE SEQUENCE FAULT SET
        ;3390 ; DEASSERT CPU FAULT BIT
        ;3391 ;  1) SEE CPU FAULT CLEAR
        ;3392 ;  2) SEE SBI FAULT CLEAR
        ;3393 ; READ NEXUS CONFIGURATION REG A AND
        ;3394 ;  SEE WRITE SEQUENCE FAULT CLEAR
        ;3395 ;
```

```
            ;3396 ;ERROR LOGOUT
            ;3397 ;
            ;3398 ; IF NONE: EITHER THE CPU FALUT OR SBI FAULT BIT IN
            ;3399 ;      ERROR(SEE LISTING)
            ;3400 ;
            ;3401 ;    DATA: EXPECTED SBI-FAULT-STATUS BITS (BITS 31-26)
            ;3402 ;      MASKED SBI-FAULT-STATUS BITS READ(BITS 31-26)
            ;3403 ;      EITHER FROM CPU OR NEXUS(SEE LISTING)
            ;3404 ;   DEVICE FLAG (SEE TEST HEADER)
            ;3405 ;   TR LEVEL OF ADAPTER UNDER TEST
            ;3406 ;   TR LEVEL OF MEMORY BEING USED
            ;3407 ;-
            ;3408 ;*****************************************************************************
            ;3409 ;
U 10E0. 0000.003D.0180.0800.0000.11D8  ;3410 =0 SUBTEST
U 10E1. 0000.003C.0180.0800.0000.10E2  ;3411  NOP
U 10E2. 0000.003D.0180.0800.0000.11B6  ;3412 =0 ERLOOP
U 10E3. 0000.003C.0180.0800.0000.10E4  ;3413  NOP
U 10E4. 0000.003D.0180.0800.0000.11F8  ;3414 =0 CALL[INIT]
U 10E5. 0800.003C.0180.0A50.0000.12F4  ;3415  D_R[0A]    ;GENERATE CACHE OFF,FORCE
U 12F4. 080D.0030.0180.0A18.0000.12F5  ;3416  D_ALU,ALU_D[OR]R[3]  ;  WRITE SEQUENCE FAULT IN D REG
U 12F5. 0000.003C.7580.3C00.0000.10E6  ;3417  ID[MAINT]_D
U 10E6. 0000.003D.1980.0800.0084.71DA  ;3418 =0 SETRCF[ZERO]
            ;3419
U 10E7. 0F10.0038.8980.0900.0284.72F6  ;3420  VA_RC[0],D_0,SC_K[.D]  ;CATCH THE TIMEOUT
U 12F6. 0801.001C.0180.0AF8.0000.12F7  ;3421  R[0F]_ALU,D_D.ORNOT.MASK
U 12F7. 0000.003C.0180.A800.0000.12F8  ;3422  MCT/WRITE.P   ;WRITE INITIATING FAULT
U 12F8. 0800.003C.0180.0A50.0000.12F9  ;3423  D_R[0A]    ;CLEAR FORCE FAULT
U 12F9. 0000.003C.7580.3C00.0000.12FA  ;3424  ID[MAINT]_D
U 12FA. 0000.003C.6DF0.2C00.0000.12FB  ;3425  Q_ID[FAULT]   ;SEE CPU FAULT SET
U 12FB. 000D.2034.0180.0A00.0010.12FC  ;3426  ALU_Q[AND]R[0],CLK.UBCC  ;TEST CPU FAULT BIT FOR SET
U 12FC. 0000.013C.0180.0800.0000.10E8  ;3427  Z?
U 10E8. 0000.003C.0180.0800.0000.12FD  ;3428 =0 J/M06470   ;CPU FAULT SET
U 10E9. 0000.003D.0180.0800.0000.11B7  ;3429  ERROR1    ;CPU FAULT BIT DID NOT SET
U 12FD. 000D.2034.0180.0A48.0010.12FE  ;3430 M06470: ALU_Q[AND]R[9],CLK.UBCC  ; CHECK THE SBI FAULT BIT
U 12FE. 0000.013C.0180.0800.0000.10EA  ;3431  Z?    ; BRANCH IF IT DID NOT SET
U 10EA. 0000.003C.0180.0800.0000.12FF  ;3432 =0 J/M06450   ; OK
U 10EB. 0000.003D.0180.0800.0000.11B7  ;3433  ERROR1    ; SBI FAULT BIT DID NOT SET
U 12FF. 0010.0038.1180.0900.0284.70EC  ;3434 M06450: SC_K[.4],VA_RC[0]
U 10EC. 0000.003D.0580.0800.0084.91DA  ;3435 =0 SC_SC+K[.1],CALL[SETRCF]
U 10ED. 0000.003C.01C0.CA38.0000.1300  ;3436  MCT/READ.P,Q_R[7]  ;READ
U 1300. 0001.003C.0180.09E8.0000.1301  ;3437  RC[0D]_D
U 1301. 0811.0034.0180.0930.0000.1302  ;3438  D_ALU,ALU_D[AND]RC[6]  ;MASK THE SBI FAULT STATUS BITS
U 1302. 001D.0020.0180.0800.0010.1303  ;3439  ALU_D[XOR]Q,CLK.UBCC  ;CHECK FOR ONLY WS FAULT
U 1303. 0001.213C.0180.09F0.0000.10EE  ;3440  Z?,RC[0E]_Q
U 10EE. 0000.003D.0180.0800.0000.11B7  ;3441 =0 ERROR1    ;WRONG NEXUS-SBI-FAULT BITS SET
U 10EF. 0800.003C.0180.0A00.0000.1304  ;3442 M06490: D_R[0]
U 1304. 0000.003C.6D80.3C00.0000.1305  ;3443  ID[FAULT]_D   ; CLEAR CPU FAULT BIT
U 1305. 0000.003C.0180.0800.0000.1306  ;3444  NOP
U 1306. 0000.003C.0180.0800.0000.1307  ;3445  NOP
U 1307. 0000.003C.0180.0800.0000.1308  ;3446  NOP
U 1308. 0F10.0038.0180.0900.0200.1309  ;3447  VA_RC[0],D_0   ;DID THEY CLEAR IN NEXUS ?
U 1309. 0000.003C.0180.C800.0000.130A  ;3448  MCT/READ.P   ;READ TO D REG
U 130A. 0811.0034.0180.0930.0000.130B  ;3449  D_ALU,ALU_D[AND]RC[6]  ;MASK FAULT BITS
U 130B. 0001.003C.0180.09E8.0010.130C  ;3450  RC[0D]_D,CLK.UBCC  ;SAVE MASKED BITS
```

```
U 130C, 0000,013C,0180,0800,0000,10F0  ;3451  Z?
U 10F0, 0000,003D,0180,0800,0000,11B7  ;3452 =0 ERROR1    ;SBI FAULT STATUS BITS SET
U 10F1, 0000,003C,0180,0800,0000,10F2  ;3453  NOP
            ;3454 ;
            ;3455 ;*****************************************************************************
            ;3456 ;+
            ;3457 ;** SUBTEST 5,6 ** UBA ONLY
            ;3458 ;
            ;3459 ; CHECKS THE INTERLOCK FUNCTION OF
            ;3460 ; NEXUS. THE
            ;3461 ; INTERLOCK FLIP-FLOP IS CHECKED TO
            ;3462 ; SEE THE  A INTERLOCK WRITE HAS TO FOLLOW
            ;3463 ; AN INTERLOCK READ.  CHECKS TO SEE THAT
            ;3464 ; AN INTERLOCK WRITE NOT PRECEEDED BY
            ;3465 ; AN INTERLOCK READ WILL CAUSE AND INTER-
            ;3466 ; LOCK SEQUENCE FAULT.
            ;3467 ;
            ;3468 ; THIS TEST SETS UP THE UBA IN WRAP AROUND MODE SO THAT
            ;3469 ; OPERATIONS TO THE SBI UNIBUS ADDRESS SPACE ARE REFERENCED
            ;3470 ; ON THE SBI ( IN MEMORY SPACE). AFTER THE MAP IS SET UP(MR0, ADDRESS
            ;3471 ; = 2000X800, WITH DATA 8000 0000) AN INTERLOCK OPERATION WILL BE MAKE
            ;3472 ; TO THE UBA ADDRESS IN I/O SPACE(SBI). THE INTERLOCK
            ;3473 ; OPERATION WILL  CAUSE THE UBA TO ISSUE AN INTERLOCK
            ;3474 ; OPERATION TO SBI MEMORY. THE MEMORY(SBI) WILL BE CHECK FOR THE
            ;3475 ; RESULT CONDITION.
            ;3476 ;
            ;3477 ;  NOTE:
            ;3478 ; (RA D) DATA1 = 0F0F0F0F
            ;3479 ; (RA E) DATA2 = F0F0F0F0
            ;3480 ;TIMEOUT = 512 BUS TICS (102.4 USEC)
            ;3481 ;
            ;3482 ;
            ;3483 ;SUBTEST 5: CHECK THE NORMAL INTERLOCK ~EQUENCE OF OPERATIONS
            ;3484 ;  I.E. INTERLOCK READ,INTERLOCK  WRITE
            ;3485 ;
            ;3486 ;SUBTEST 6: CHECK THE INTERLOCKLOCK SEQUENCE FAULT
            ;3487 ;  I.E. A INTERLOCK WRITE BEFORE AN INTERLOCK READ
            ;3488 ;-
            ;3489 ;*****************************************************************************
            ;3490 ;+
            ;3491 ; SUBTEST 5 UBA ONLY
            ;3492 ;
            ;3493 ; CHECK NORMAL INTERLOCK SEQUENCE
            ;3494 ; I.E. INTERLOCK READ,INTERLOCK WRITE
            ;3495 ;
            ;3496 ;
            ;3497 ;
            ;3498 ;ERROR LOGOUT
            ;3499 ;
            ;3500 ; DATA:   SEE THE "NOINTR" ROUTINE
            ;3501 ;-
            ;3502 ;*****************************************************************************
            ;3503 ;
U 10F2, 0000,003D,0180,0800,0000,11D8  ;3504 =0 SUBTEST
U 10F3, 0000,003C,0180,0800,0000,10F4  ;3505  nop
```

```
U 10F4. 0000.003D.0180.0800.0000.11DD  ;3506 =0 delay
U 10F5. 0010.0038.0180.0960.0010.130D  ;3507  ALU_RC[0C],CLK.UBCC
U 130D. 0000.013C.0180.0800.0000.10F6  ;3508  Z?
U 10F6. 0000.003C.0180.0800.0000.1116  ;3509 =0 J/M06600   ;SKIP THIS SUBTEST
U 10F7. 0000.003C.0180.0800.0000.10F8  ;3510  NOP
U 10F8. 0000.003D.0180.0800.0000.11B6  ;3511 =0 ERLOOP
U 10F9. 0000.003C.0180.0800.0000.10FA  ;3512  NOP
U 10FA. 0000.003D.0180.0800.0000.11F8  ;3513 =0 CALL[INIT]
U 10FB. 0000.003C.0180.0800.0000.10FC  ;3514  NOP
U 10FC. 0000.003D.0180.0800.0000.1225  ;3515 =0 CALL[WAITFORUNIBUS]  ; WAIT FOR UNIBUS TO COME UP
U 10FD. 0000.003C.0180.0800.0000.10FE  ;3516  NOP
U 10FE. 0000.003D.1980.0800.0084.71DA  ;3517 =0 SETRCF[ZERO]
U 10FF. 0810.0038.8580.0900.0084.730E  ;3518  D_RC[0],SC_K[.C]  ;SET UP MAP REGISTER 0( MR0)
U 130E. 0000.003C.0580.0800.0084.B30F  ;3519  SC_SC-K[.1]
U 130F. 0801.001C.0180.0800.0200.1310  ;3520  VA_ALU,D_D.ORNOT.MASK  ;SET ADDRESS
U 1310. 0F00.003C.8D80.0800.0084.7311  ;3521  D_0,SC_K[.1F]   ;GENERATE THE DATA
U 1311. 0801.001C.0180.0800.0000.1312  ;3522  D_D.ORNOT.MASK
U 1312. 0000.003C.0180.A800.0000.1313  ;3523  MCT/WRITE.P
U 1313. 0F10.0038.8980.0938.0284.7314  ;3524  VA_RC[7],SC_K[.D],D_0  ; SET ADDRESS FOR INTERLOCK READ
U 1314. 0801.001C.0180.0AF8.0000.1315  ;3525  R[0F]_ALU,D_D.ORNOT.MASK ;SET U-TRAP FLAG
U 1315. 0000.403C.0180.E800.0000.1316  ;3526  MCT/LOCKREAD.P,WORD
U 1316. 0000.003C.0180.0A78.0010.1317  ;3527  ALU_R[0F],CLK.UBCC  ;CHECK TIME OUT FLAG
U 1317. 0000.013C.0180.0800.0000.110A  ;3528  Z?
U 110A. 0000.003C.0180.0800.0000.1040  ;3529 =0 J/M06500   ;NO TIMEOUT
U 110B. 0000.003D.0180.0800.0000.11B7  ;3530  ERROR1    ;INTERLOCK READ TIMED OUT
        ;3531 =00
U 1040. 0000.003D.0180.0800.0000.11DF  ;3532 M06500: CALL[NOINTR]
U 1041. 0000.003D.0180.0800.0000.11B7  ;3533  ERROR1    ;UNEXPECTED INTERRUPT !!!!!
U 1042. 0000.403C.0180.B800.0000.1043  ;3534  MCT/LOCKWRITE.P,WORD  ;INTERLOCK WRITE
U 1043. 0000.003C.0180.0800.0000.1110  ;3535  NOP
        ;3536 =0 D_ALU.LEFT,ALU_K[.FF],SI/MUL-,
U 1110. 0838.0039.4B80.0800.0000.11DE  ;3537   CALL[WAIT.LOOP]
U 1111. 0000.003C.0180.0800.0000.1044  ;3538  NOP
U 1044. 0000.003D.0180.0800.0000.11DF  ;3539 =00 CALL[NOINTR]
U 1045. 0000.003D.0180.0800.0000.11B7  ;3540  ERROR1    ;INTERLOCK WRITE PROBLEM
        ;3541       ;UNEXPECTED INTERRUPT !!!!!
U 1046. 0F10.0038.8980.0938.0284.7047  ;3542  D_0,SC_K[.D],VA_RC[7]  ;CATCH A TIMEOUT ON THE INTERLOCK
U 1047. 0801.001C.0180.0AF8.0000.1318  ;3543  R[0F]_ALU,D_D.ORNOT.MASK ;  READ,CHECK TO SEE INTERLOCK
        ;3544       ;   WRITE RESET THE INTERLOCK
        ;3545       ;    FLIP FLOP IN NEXUS
U 1318. 0000.403C.0180.E800.0000.1319  ;3546  MCT/LOCKREAD.P,WORD  ;INTERLOCK READ
U 1319. 0000.003C.0180.0A78.0010.131A  ;3547  ALU_R[0F],CLK.UBCC  ;CHECK THE TIMEOUT
U 131A. 0000.013C.0180.0800.0000.1112  ;3548  Z?
U 1112. 0000.003C.0180.0800.0000.1114  ;3549 =0 J/M06550   ;NO TIME OUT
U 1113. 0000.003D.0180.0800.0000.11B7  ;3550  ERROR1    ;INTERLOCK READ TIMED OUT
        ;3551       ;  NEXUS INTERLOCK FLOP NOT RESET
        ;3552 =0
        ;3553 M06550: D_ALU.LEFT,SI/MUL-,ALU_K[.FF], ;
U 1114. 0838.0039.4B80.0800.0000.11DE  ;3554   CALL[WAIT.LOOP]  ;LET NEXUS INTERLOCK FLOP RESET
U 1115. 0000.003C.0180.0800.0000.1116  ;3555  NOP
        ;3556 ;
        ;3557 ;*************************************************************************
        ;3558 ;*
        ;3559 ;** SUBTEST 6 ** UBA ONLY
        ;3560 ;
```

```
            ;3561 ; WRITE INTERLOCK, WITH NO PIROR INTERLOCK READ
            ;3562 ; SEE A TIMEOUT DUE TO NEXUS RESPONDING WITH
            ;3563 ; BUSY. CHECK CPU SBI-FAULT-STATUS REGISTER TO
            ;3564 ; SEE THAT IT RECORDED ITSELF AS TRANSMITTER.
            ;3565 ; READ THE NEXUS SBI-FAULT-STATUS TO SEE THAT
            ;3566 ; IT DETECTED THE INTERLOCK SEQUENCE FAULT.
            ;3567 ;
            ;3568 ; NOTE : THE UBA MAP SHOULD STILL BE SET UP
            ;3569 ;
            ;3570 ;ERROR LOGOUT
            ;3571 ;
            ;3572 ; IF NONE: EITHER THE CPU FALUT OR SBI FAULT BIT IN
            ;3573 ;       ERROR(SEE LISTING)
            ;3574 ;
            ;3575 ;    DATA: EXPECTED SBI-FAULT-STATUS BITS (BITS 31-26)
            ;3576 ;      MASKED SBI-FAULT-STATUS BITS READ(BITS 31-26)
            ;3577 ;      EITHER FROM CPU OR NEXUS(SEE LISTING)
            ;3578 ;   DEVICE FLAG (SEE TEST HEADER)
            ;3579 ;   TR LEVEL OF ADAPTER UNDER TEST
            ;3580 ;   TR LEVEL OF MEMORY BEING USED
            ;3581 ;-
            ;3582 ;
            ;3583 ;*****************************************************************************
            ;3584 ;
            ;3585 =0
U 1116, 0000,003D,0180,0800,0000,11D8   ;3586 M06600: SUBTEST
U 1117, 0000,003C,0180,0800,0000,1118   ;3587  NOP
U 111B, 0000,003D,0180,0800,0000,11DD   ;3588 =0 DELAY
U 1119, 0010,0038,0180,0960,0010,131B   ;3589  ALU_RC[0C],CLK.UBCC
U 131B, 0000,013C,0180,0800,0000,111A   ;3590  Z?
U 111A, 0000,003C,0180,0800,0000,113C   ;3591 =0 J/M06800   ;SKIP THIS SUBTEST
U 111B, 0000,003C,0180,0800,0000,111C   ;3592  NOP
U 111C, 0000,003D,0180,0800,0000,11B6   ;3593 =0 ERLOOP
U 111D, 0000,003C,0180,0800,0000,111E   ;3594  NOP
U 111E, 0000,003D,0180,0800,0000,11F8   ;3595 =0 CALL[INIT]
U 111F, 0000,003C,0180,0800,0000,1120   ;3596  NOP
U 1120, 0000,003D,0180,0800,0000,1225   ;3597 =0 CALL[WAITFORUNIBUS]  ; WAIT FOR UNIBUS
U 1121, 0010,0038,0180,0938,0200,1122   ;3598  VA_RC[7]
            ;3599 =0 D_ALU.LEFT,ALU_K[.FF],SI/MUL-,
U 1122, 0838,0039,4B80,0800,0000,11DE   ;3600   CALL[WAIT.LOOP]
U 1123, 0000,003C,0180,0800,0000,1124   ;3601  NOP
U 1124, 0000,003D,1980,0800,0084,71DA   ;3602 =0 SETRCF[ZERO]
U 1125, 0F00,003C,8980,0800,0084,731C   ;3603  D_0,SC_K[.D]   ;CATCH THE TIMEOUT U-TRAP
U 131C, 0001,001C,0180,0AF8,0000,131D   ;3604  R[0F]_ALU,ALU_D.ORNOT.MASK
U 131D, 0000,403C,0180,B800,0000,1126   ;3605  MCT/LOCKWRITE.P,WORD  ;INITIATE WRITE OPERATION
            ;3606 =0 D_ALU.LEFT,ALU_K[.FF],SI/MUL-,
U 1126, 0838,0039,4B80,0800,0000,11DE   ;3607   CALL[WAIT.LOOP]
U 1127, 0000,003C,0180,0800,0000,1048   ;3608  NOP
U 104B, 0000,003D,0180,0800,0000,1202   ;3609 =00 CALL[WRTOUT]
U 1049, 0000,003D,0180,0800,0000,11B7   ;3610  ERROR1    ;WRONG INTERRUPT TAKEN
U 104A, 0000,003C,8580,0800,0084,704B   ;3611  SC_K[.C]   ; CLEAR THE TIMEOUT BIT
U 104B, 0803,001C,0180,0800,0000,131E   ;3612  D_NOT.MASK
U 131E, 0000,003C,6580,3C00,0000,131F   ;3613  ID[SBI.ERR]_D
U 131F, 0000,003C,6DF0,2C00,0000,1320   ;3614  Q_ID[FAULT]   ;CPU FAULT SET ??
U 1320, 000D,2034,0180,0A00,0010,1321   ;3615  ALU_Q[AND]R[0],CLK.UBCC  ;TEST CPU FAULT BIT
```

```
U 1321. 0000.013C.0180.0800.0000.1128  ;3616  Z?
U 1128. 0000.003C.0180.0800.0000.1322  ;3617 =0 J/M06670   ;CPU FAULT SET
U 1129. 0000.003D.0180.0800.0000.11B7  ;3618  ERROR1    ;CPU FAULT BIT NOT SET
U 1322. 000D.2034.0180.0A48.0010.1323  ;3619 M06670: ALU_Q[AND]R[09],CLK.UBCC ;TEST SBI FAULT BIT
U 1323. 0000.013C.0180.0800.0000.112A  ;3620  Z?
U 112A. 0000.003C.0180.0800.0000.1324  ;3621 =0 J/M06680   ;BIT WAS SET
U 112B. 0000.003D.0180.0800.0000.11B7  ;3622  ERROR1    ;SBI FAULT BIT DID NOT SET
U 1324. 0011.2034.01C0.0930.0000.1325  ;3623 M06680: Q_ALU,ALU_Q[AND]RC[6]  ;MASK SBI STATUS BITS IN CPU
U 1325. 0001.203C.0180.09E8.0000.1326  ;3624  RC[0D]_Q   ;SAVE THEM
U 1326. 000D.2020.0180.0A58.0010.1327  ;3625  ALU_Q[XOR]R[0B],CLK.UBCC ;DID THE CPU RECORD ITSELF
        ;3626      ;  AS TRANSMITTER DURING FAULT ?
U 1327. 0800.003C.1180.0A58.0084.712C  ;3627  SC_K[.4],D_R[0B]
U 112C. 0000.003D.0580.0800.0084.91DA  ;3628 =0 SC_SC+K[.1],CALL[SETRCF]
U 112D. 0001.013C.0180.09F0.0000.112E  ;3629  Z?,RC[0E]_D
U 112E. 0000.003D.0180.0800.0000.11B7  ;3630 =0 ERROR1    ;CPU DID NOT RECORD ITSELF
        ;3631      ;    AS TRANSMITTER DURING FAULT
U 112F. 0010.0038.0180.0900.0200.1328  ;3632  VA_RC[0]   ; READ NEXUS FAULT STATUS
U 1328. 0000.003C.0180.C800.0000.1329  ;3633  MCT/READ.P
U 1329. 0811.0034.0180.0930.0000.132A  ;3634  D_ALU,ALU_D[AND]RC[6]  ;MASK
U 132A. 0001.003C.0180.09E8.0000.132B  ;3635  RC[0D]_D
U 132B. 0011.0020.0180.0928.0010.132C  ;3636  ALU_D[XOR]RC[5],CLK.UBCC ;CHECK THE SBI-FAULT BITS SET
U 132C. 0010.0038.01C0.0928.0000.132D  ;3637  Q_RC[5]
U 132D. 0001.213C.0180.09F0.0000.1130  ;3638  Z?,RC[0E]_Q
U 1130. 0000.003D.0180.0800.0000.11B7  ;3639 =0 ERROR1    ;WRONG BITS SET
U 1131. 0800.003C.0180.0A00.0000.1132  ;3640  D_R[0]
U 1132. 0000.003C.6D80.3C00.0000.1133  ;3641 =0 ID[FAULT]_D   ;CLEAR CPU FAULT
U 1133. 0000.003C.0180.0800.0000.1134  ;3642  NOP
U 1134. 0000.003D.1980.0800.0084.71DA  ;3643 =0 SETRCF[ZERO]
U 1135. 0000.003C.0180.0800.0000.132E  ;3644  NOP
U 132E. 0000.003C.6DF0.2C00.0000.132F  ;3645  Q_ID[FAULT]
U 132F. 000D.2034.0180.0A48.0010.1330  ;3646  ALU_Q[AND]R[9],CLK.UBCC  ;SBI FAULT CLEAR  ??
U 1330. 0000.013C.0180.0800.0000.1136  ;3647  Z?
U 1136. 0000.003D.0180.0800.0000.11B7  ;3648 =0 ERROR1    ;SBI FAULT BIT SET
U 1137. 0010.0038.0180.0900.0200.1331  ;3649  VA_RC[0]   ;READ NEXUS CONFIG REG A FOR SBI-FAULT-STATUS
U 1331. 0000.003C.1180.0800.0084.7138  ;3650  SC_K[.4]
U 1138. 0000.003D.0580.0800.0084.91DA  ;3651 =0 SC_SC+K[.1],CALL[SETRCF]
U 1139. 0003.003C.0180.C9F0.0000.1332  ;3652  MCT/READ.P,RC[0E]_0  ;READ NEXUS SBI-STATUS BITS
U 1332. 0811.0034.0180.0930.0000.1333  ;3653  D_ALU,ALU_D[AND]RC[6]  ;MASK FAULT BITS
U 1333. 0001.003C.0180.09E8.0000.1050  ;3654  RC[0D]_D   ;SAVE MASKED BITS
U 1050. 0000.003D.0180.0800.0000.11DF  ;3655 =00 CALL[NOINTR]
U 1051. 0000.003D.0180.0800.0000.11B7  ;3656  ERROR1    ;UNEXPECTED INTERRUPT !!!!!
U 1052. 0010.0038.0180.0968.0010.1334  ;3657  ALU_RC[0D],CLK.UBCC
U 1334. 0000.013C.0180.0800.0000.113A  ;3658 = Z?
U 113A. 0000.003D.0180.0800.0000.11B7  ;3659 =0 ERROR1    ;SBI FAULT STATUS BITS SET
U 113B. 0000.003C.0180.0800.0000.113C  ;3660  NOP
        ;3661 ;
        ;3662 ;*************************************************************************
        ;3663 ;*
        ;3664 ;** SUBTEST 7 **  UBA ONLY
        ;3665 ;
        ;3666 ;INTERLOCK TIMING TESTS
        ;3667 ;
        ;3668 ; FIRST CHECK THE INTERLOCK READ,INTERLOCK READ SEQUENCE
        ;3669 ; THIS WILL ASSURE THE INTERLOCK TIME COUNTER IN
        ;3670 ; NEXUS CONTROLLER COUNT LESS THEN 512 TICKS(APPROX)
```

```
            ;3671  ; I.E. NEXUS COUNTER SHOULD FINISH BEFORE THE CPU TIMES
            ;3672  ; OUT
            ;3673  ;
            ;3674  ; ERROR LOGOUT
            ;3675  ;
            ;3676  ; DATA: NOT RELEVENT
            ;3677  ;  NOT RELEVENT
            ;3678  ;  DEVICE FLAG (SEE TEST HEADER)
            ;3679  ;  TR LEVEL OF ADAPTER UNDER TEST
            ;3680  ;  TR LEVEL OF MEMORY BEING USED
            ;3681  ;
            ;3682  ; NOTE: SEE LISTING FOR FAILING EVENT
            ;3683  ;
            ;3684  ;-
            ;3685  ;*****************************************************************************
            ;3686  ;
            ;3687  =0
U 113C, 0000,003D,0180,0800,0000,11D8   ;3688 M06800: SUBTEST
            ;3689  ;
            ;3690  ;
            ;3691  ; SKIP THIS SUBTEST
            ;3692  ;
U 113D, 0000,003C,0180,0800,0000,114C   ;3693  J/M06815
U 113E, 0000,003D,0180,0800,0000,11DD   ;3694 =0 DELAY
U 113F, 0010,0038,0180,0960,0010,1335   ;3695  ALU_RC[0C],CLK.UBCC
U 1335, 0000,013C,0180,0800,0000,1140   ;3696  Z?
U 1140, 0000,003C,0180,0800,0000,114C   ;3697 =0 J/M06815    ;SKIP THIS SUBTEST
U 1141, 0000,003C,0180,0800,0000,1142   ;3698  NOP
U 1142, 0000,003D,0180,0800,0000,11B6   ;3699 =0 ERLOOP
U 1143, 0000,003C,0180,0800,0000,1144   ;3700  NOP
U 1144, 0000,003D,0180,0800,0000,11F8   ;3701 =0 CALL[INIT]
U 1145, 0000,003C,0180,0800,0000,1146   ;3702  NOP
U 1146, 0000,003D,0180,0800,0000,1225   ;3703 =0 CALL[WAITFORUNIBUS]  ; WAIT FOR THE UNIBUS
U 1147, 0000,003C,1180,0800,0084,7148   ;3704  SC_K[.4]
U 1148, 0000,003D,0580,0800,0084,91DA   ;3705 =0 SC_SC+K[.1],CALL[SETRCF]
U 1149, 0010,0038,0180,0938,0200,1336   ;3706  VA_RC[7]
U 1336, 0000,403C,0180,E800,0000,1054   ;3707 = MCT/LOCKREAD.P,WORD  ;START THE NEXUS COUNTER
U 1054, 0000,003D,0180,0800,0000,11DF   ;3708 =00 CALL[NOINTR]
U 1055, 0000,003D,0180,0800,0000,11B7   ;3709  ERROR1     ;UNEXPECTED INTERRUPT !!!!!
U 1056, 0F00,003C,8980,0800,0084,7337   ;3710  D_0,SC_K[.D]    ;CATCH A TIMEOUT IF OCCURES
U 1337, 0801,001C,0180,0AF8,0000,1338   ;3711 = R[0F]_ALU,D_D.ORNOT.MASK
U 1338, 0000,403C,0180,E800,0000,1339   ;3712  MCT/LOCKREAD.P,WORD  ;START 2ND READ
U 1339, 0000,003C,0180,0A78,0010,133A   ;3713  ALU_R[0F],CLK.UBCC  ;SEE NO TIMEOUT,I.E. NEXUS
            ;3714       ;TIMER SHOULD RESET BEFORE
            ;3715       ;  CPU TIMES-OUT
U 133A, 0000,013C,0180,0800,0000,114A   ;3716  Z?
U 114A, 0000,003C,0180,0800,0000,1060   ;3717 =0 J/M06810    ;NO TIMEOUT OCCURED
U 114B, 0000,003D,0180,0800,0000,11B7   ;3718  ERROR1     ;TIMEOUT OCCURED,COUNTER IN
            ;3719       ; NEXUS COUNTED TO LONG
            ;3720 =00
U 1060, 0000,003D,0180,0800,0000,11DF   ;3721 M06810: CALL[NOINTR]
U 1061, 0000,003D,0180,0800,0000,11B7   ;3722  ERROR1     ;SOME OTHER ERROR OCCURED
            ;3723       ;UNEXPECTED INTERRUPT  !!!!!!
U 1062, 0000,003C,0180,0800,0000,133B   ;3724  NOP
U 133B, 0003,003C,0180,0AF8,0000,114C   ;3725 = R[0F]_0     ;CLEAR U-TRAP FLAG
```

```
                ;3726 ;
                ;3727 ;=============================================================================
                ;3728 ;*
                ;3729 ; SUBTEST 8 UNEXPECTED READ DATA FAULT
                ;3730 ;
                ;3731 ;  UNEXPECTED READ DATA FAULT TO
                ;3732 ;  NEXUS. CHECK TO SEE THAT THE NEXUS SBI
                ;3733 ;  INTERFACE LOGIC CAN DETECT UNEXPECTED
                ;3734 ;  READ DATA FAULTS.  THE NEXUS ID
                ;3735 ;  WILL BE SET AS THE MAINTENANCE ID AND
                ;3736 ;  THE UNEXPECTED READ DATA BIT (ID=1D,#29) WILL
                ;3737 ;  BE SET.  THE SBI FAULT INTERRUPT WILL BE
                ;3738 ;  CHECKED AS WELL AS THE UNEXPECTED READ DATA
                ;3739 ;  BIT IN THE FAULT (BIT #28 ID=13) REGISTER.
                ;3740 ;  THE CACHE WILL BE DISABLED.
                ;3741 ;
                ;3742 ; ERROR LOGOUT
                ;3743 ;
                ;3744 ; IF NO PRINT OUT INDIVIDUAL BIT(NEXUS,SBI FAULT BIT)IN ERROR
                ;3745 ;
                ;3746 ; DATA: EXPECTED DATA
                ;3747 ;  MASKED IMAGE OF SBI-FAULT-STATUS REGISTER
                ;3748 ;  DEVICE FLAG (SEE TEST HEADER)
                ;3749 ;  TR LEVEL OF ADAPTER UNDER TEST
                ;3750 ;  TR LEVEL OF MEMORY BEING USED
                ;3751 ;
                ;3752 ;-
                ;3753 ;=============================================================================
                ;3754 ;
                ;3755 ;
                ;3756 =0
U 114C. 0000.003D.0180.0800.0000.11D8  ;3757 M06815: SUBTEST
U 114D. 0000.003C.0180.0800.0000.114E  ;3758  NOP
U 114E. 0000.003D.0180.0800.0000.11B6  ;3759 =0 ERLOOP
U 114F. 0000.003C.0180.0800.0000.1150  ;3760  NOP
U 1150. 0000.003D.0180.0800.0000.11F8  ;3761 =0 CALL[INIT]
U 1151. 0000.003C.7DF8.0800.0084.733C  ;3762  SC_K[.18],Q_0    ;SHIFT ID TO
U 133C. 0810.0038.0580.0958.0084.B33D  ;3763  SC_SC-K[.1],D_RC[0B]  ; MAINTENANCE ID
U 133D. 0D00.003C.0180.0800.0000.133E  ;3764  D_DAL.SC    ; POSITION
U 133E. 0000.003C.75F0.2C00.0000.133F  ;3765  Q_ID[MAINT]    ;GET CACHE OFF BITS
U 133F. 081D.0030.ED80.0800.0084.7340  ;3766  D_D[OR]Q,SC_K[.1B]  ;SET FORCE
U 1340. 0000.003C.0980.0800.0084.9341  ;3767  SC_SC+K[.2]    ;  UNEXPECTED READ DATA
U 1341. 0801.001C.0180.0800.0000.1342  ;3768  D_D.ORNOT.MASK
U 1342. 0C00.003C.7580.3C00.0000.1343  ;3769  ID[MAINT]_D,D_Q    ;FORCE FAULT
U 1343. 0000.003C.7580.3C00.0000.1344  ;3770  ID[MAINT]_D
U 1344. 0000.003C.1180.0800.0084.7152  ;3771  SC_K[.4]
U 1152. 0000.003D.0580.0800.0084.91DA  ;3772 =0 SC_SC+K[.1],CALL[SETRCF]
U 1153. 0000.003C.0180.0800.0000.1064  ;3773  NOP     ;READ FAULT
U 1064. 0000.003D.0180.0800.0000.11FB  ;3774 =00 CALL[SBIFLT]    ;SEE SBI FAULT
U 1065. 0000.003D.0180.0800.0000.11B7  ;3775  ERROR1     ;DIDN'T GET SBI FAULT INTERRUPT !!
U 1066. 0000.003C.6DF0.2C00.0000.1067  ;3776  Q_ID[FAULT]
U 1067. 081B.0038.4580.0800.0000.1345  ;3777  D_K[.8000]
U 1345. 0101.203C.0180.09E8.0000.1346  ;3778  D_D.LEFT2,RC[0D]_Q  ;SEE SBI FAULT SET
U 1346. 001D.0034.0180.0800.0010.1347  ;3779  ALU_D[AND]Q,CLK.UBCC
U 1347. 0000.013C.0180.0800.0000.1154  ;3780  Z?
```

```
U 1154. 0000.003C.0180.0800.0000.1348  ;3781 =0 J/M19E00
U 1155. 0000.003D.0180.0800.0000.11B7  ;3782  ERROR1     ;SBI FAULT NOT SET
U 1348. 0100.003C.0180.0800.0000.1349  ;3783 M19E00: D_D.LEFT2    ;CHECK  CPU FAULT
U 1349. 001D.0034.0180.0800.0010.134A  ;3784  ALU_D[AND]Q,CLK.UBCC  ;  SET
U 134A. 0000.013C.0180.0800.0000.1156  ;3785  Z?
U 1156. 0000.003C.0180.0800.0000.134C  ;3786 =0 J/M19E01
U 1157. 0000.003D.0180.0800.0000.11B7  ;3787  ERROR1     ;CPU FAULT NOT SET
U 134B. 0000.003C.0180.0800.0000.134C  ;3788  NOP
U 134C. 0010.0038.0180.0900.0200.134D  ;3789 M19E01: VA_RC[0]    ; READ THE NEXUS STATUS REGISTER
U 134D. 0000.003C.0180.C800.0000.134E  ;3790  D[LONG]_CACHE.P
U 134E. 0811.0034.0180.0930.0000.134F  ;3791  D_D.AND.RC[6]    ; MASK THE FAULT BITS
U 134F. 0010.0038.01C0.0940.0000.1350  ;3792  Q_RC[8]     ; GET EXPECTED DATA
U 1350. 001D.2000.0180.0800.0010.1351  ;3793  ALU_Q-D,CLK.UBCC  ; CHECK RECEIVED FAULT BITS
U 1351. 0001.213C.0180.09F0.0000.1158  ;3794  Z?,RC[0E]_Q    ; BRANCH IF OK
U 1158. 0001.003D.0180.09E8.0000.11B7  ;3795 =0 ERROR1,RC[0D]_D    ; BIT 29 IN NEXUS NOT SET
          ;3796       ;=======================================
          ;3797       ;DATA: EXPECTED DATA
          ;3798       ; MASK IMAGE NEXUS-FAULT REG
          ;3799       ;=======================================
U 1159. 0000.003C.0180.0800.0000.115A  ;3800  NOP
U 115A. 0000.003D.0180.0800.0000.1213  ;3801 =0 CALL[CLRFLT]    ;CLEAR CPU FAULT
U 115B. 0818.0038.4580.0800.0000.1352  ;3802  D_K[.8000]
U 1352. 0100.003C.0180.0800.0000.1353  ;3803  D_D.LEFT2
U 1353. 0100.003C.6DF0.2C00.0000.1354  ;3804  Q_ID[FAULT],D_D.LEFT2
U 1354. 001D.0034.0180.0800.0010.1355  ;3805  ALU_D[AND]Q,CLK.UBCC  ;CPU FAULT CLEAR?
U 1355. 0000.013C.0180.0800.0000.115C  ;3806  Z?
U 115C. 0000.003D.0180.0800.0000.11B7  ;3807 =0 ERROR1     ;CPU FAULT DID NOT CLEAR
U 115D. 0200.003C.0180.0800.0000.1356  ;3808  D_D.RIGHT2
U 1356. 001D.0034.0180.0800.0010.1357  ;3809  ALU_D[AND]Q,CLK.UBCC
U 1357. 0000.013C.0180.0800.0000.115E  ;3810  Z?     ;SBI FAULT CLEAR?
U 115E. 0000.003D.0180.0800.0000.11B7  ;3811 =0 ERROR1     ;NO, SBI FAULT DID NOT CLEAR
U 115F. 0010.0038.0180.0900.0200.1358  ;3812  VA_RC[0]    ; READ THE NEXUS STATUS REGISTER
U 1358. 0003.003C.0180.C9F0.0000.1359  ;3813  D[LONG]_CACHE.P,RC[0E]_0 ; ...
U 1359. 0811.0034.0180.0930.0010.135A  ;3814  D_D.AND.RC[6],CLK.UBCC  ; MASK AND CHECK
U 135A. 0000.013C.0180.0800.0000.1160  ;3815  Z?
U 1160. 0001.003D.0180.09E8.0000.11B7  ;3816 =0 ERROR1,RC[0D]_D    ; FAULT BIT IN NEXUS DID NOT CLEAR
          ;3817       ;=======================================
          ;3818       ;DATA: EXPECTED DATA
          ;3819       ; MASKED IMAGE OF NEXUS-
          ;3820       ;      SBI-STATUS REGISTER
          ;3821       ;=======================================
U 1161. 0000.003C.0180.0800.0000.1162  ;3822  NOP
          ;3823 ;
          ;3824 ;================================================================================
          ;3825 ;*
          ;3826 ; SUBTEST 9 MULTIPLE TRANSMITTER TEST FOR NEXUS
          ;3827 ;
          ;3828 ;    THIS TEST FORCE ALL 32 I.D. CODES ON THE
          ;3829 ; BUS AND CHECKS THAT NEXUS RESPONDS PROPERLY. SBI PARITY
          ;3830 ; WILL BE MASKED OFF SINCE THE FORCING OF MULTIPLE XMITERS
          ;3831 ; ERRORS CAUSE BAD PARITY ON THE BUS. THE BUS FUNCTIONALY
          ;3832 ; OR'S TOGETHER THE I.D. BITS ACTIVE. THOSE MAINTENAINCE I.D.'S
          ;3833 ; WHICH WHEN OR'D WITH THE CPU I.D. DO NOT CHANGE THE CPU I.D.
          ;3834 ; WILL NOT CAUSE MULTIPLE TRANSMITTER FAULTS.
          ;3835 ;
```

```
            ;3836 ; ERROR LOGOUT
            ;3837 ;
            ;3838 ; DATA: RECEIVED FAULT BITS
            ;3839 ;  I.D. UNDER TEST (2ND NON-ERROR RESPONSE)
            ;3840 ;  DEVICE FLAG (SEE TEST HEADER)
            ;3841 ;  TR LEVEL OF ADAPTER UNDER TEST
            ;3842 ;  TR LEVEL OF MEMORY BEING USED
            ;3843 ;
            ;3844 ;
            ;3845 ; NOTE: SEE LISTING FOR FAILNG EVENTS AND
            ;3846 ;         SPECIFIC ERROR OCCURENCE
            ;3847 ;
            ;3848 ;-
            ;3849 ;=============================================================================
            ;3850 ;
U 1162. 0000.003D.0180.0800.0000.11D8  ;3851 =0 SUBTEST
U 1163. 0003.003C.0180.0800.0200.135B  ;3852 SBE005: VA_ALU,ALU_0(A)   ; FIND THE CPU ID
U 135B. 0818.0038.7580.0800.0000.135C  ;3853  D_K[.20]
U 135C. 0B00.003C.0180.0800.0000.135D  ;3854  D_D.SWAP
U 135D. 0000.003C.7180.3C00.0000.135E  ;3855  ID[COMP]_D   ; START THE SILO COUNTER
U 135E. 0000.003C.0180.C800.0000.1164  ;3856  MCT/READ.P   ; START AN SBI CYCLE
            ;3857 =0 D_K[.14],
U 1164. 0818.0039.2180.0800.0000.11DE  ;3858   CALL[WAIT.LOOP]
            ;3859 KEEPLOOKING:
U 1165. 0000.003C.61F0.2C00.0000.135F  ;3860  Q_ID[SILO]   ; SEARCH SILO FOR FIRST NON ZERO ENTRY
U 135F. 0C01.203C.7DB0.0800.0010.1360  ;3861  ALU_Q,CLK.UBCC,K[.18],D_Q
U 1360. 001B.0100.7D80.0800.0082.1166  ;3862  Z?,SC_0-K[.18]
U 1166. 0000.003C.0180.0800.0000.1361  ;3863 =0 J/FOUNDIT
U 1167. 0000.003C.0180.0800.0000.1165  ;3864  J/KEEPLOOKING
            ;3865 FOUNDIT:
U 1361. 0000.003C.05F8.0800.0084.B362  ;3866  SC_SC-K[.1],Q_0   ; SETUP TO SHIFT ID INTO BITS<4:0>
U 1362. 0D00.003C.0180.0800.0000.1363  ;3867  D_DAL.SC
U 1363. 0019.0034.8D80.09C8.0000.1168  ;3868  RC[9]_D.AND.K[.1F]  ; MASK AND SAVE CPU ID
U 1168. 0000.003D.0180.0800.0000.11B6  ;3869 =0 ERLOOP
U 1169. 0003.003C.0180.09E8.0000.1364  ;3870  RC[0D]_0   ;SET I.D. COUNTER
U 1364. 0000.003C.0180.0800.0000.116A  ;3871 SBE010: NOP
U 116A. 0000.003D.0180.0800.0000.11F8  ;3872 =0 CALL[INIT]
U 116B. 0F00.003C.7D80.0800.0084.7365  ;3873  SC_K[.18],D_0   ;GENERATE CACHE OFF,
U 1365. 0000.003C.1180.0800.0084.9366  ;3874  SC_SC+K[.4]   ;     AND FORCE MX IMAGE
U 1366. 0801.001C.75F0.2C00.0000.1367  ;3875  D_D.ORNOT.MASK,Q_ID[MAINT]
U 1367. 081D.0030.1180.0800.0084.716C  ;3876  SC_K[.4],D_D[OR]Q
U 116C. 0000.003D.0580.0800.0084.91DA  ;3877 =0 SC_SC+K[.1],CALL[SETRCF]
U 116D. 0000.003C.7580.3C00.0000.1368  ;3878  ID[MAINT]_D
U 1368. 0810.0038.21F8.0968.0084.7369  ;3879  SC_K[.14],Q_0,D_RC[0D]  ;SHIFT ID COUNT TO
U 1369. 0000.003C.0D80.0800.0084.936A  ;3880  SC_SC+K[.3]   ;     MAINT ID POSITION
U 136A. 0D00.003C.0180.0800.0000.136B  ;3881  D_DAL.SC
U 136B. 0000.003C.75F0.2C00.0000.136C  ;3882  Q_ID[MAINT]   ;GET CACHE OFF, FORCE MX
U 136C. 081D.0030.0180.0800.0000.136D  ;3883  D_D[OR]Q
U 136D. 0000.003C.7580.3C00.0000.136E  ;3884  ID[MAINT]_D
U 136E. 0F00.003C.8980.0800.0084.736F  ;3885  D_0,SC_K[.D]
U 136F. 0010.0038.0180.0900.0200.1370  ;3886  VA_RC[0]   ;CONFIG REG ADDRESS
U 1370. 0801.001C.0180.0AF8.0000.1371  ;3887  R[0F]_ALU,D_D.ORNOT.MASK ;CATCH THE TIMEOUT
U 1371. 0000.003C.0180.C800.0000.1372  ;3888  MCT/READ.P   ;READ MAKING NEXUS
U 1372. 0F00.003C.0180.0800.0000.1373  ;3889  D_0
U 1373. 0000.003C.7580.3C00.0000.1374  ;3890  ID[MAINT]_D
```

```
U 1374. 0010.0038.0180.0900.0200.1375  ;3891  VA_RC[0]   ;READ NEXUS CONFIG REG A
U 1375. 0000.003C.0180.C800.0000.1376  ;3892  MCT/READ.P
U 1376. 0011.0034.01C0.0930.0000.1377  ;3893  Q_D.AND.RC[6]   ; MASK FAULT BITS
U 1377. 0001.203C.0180.0AF0.0000.1378  ;3894  R[0E]_Q
U 1378. 0010.0038.01C0.0968.0000.1379  ;3895  Q_RC[0D]   ;GET CURRENT ID
U 1379. 0810.0038.0180.0948.0000.137A  ;3896  D_RC[9]    ; GET CPU ID
U 137A. 001D.0030.01C0.0800.0000.137B  ;3897  Q_D[OR]Q   ; OR TEST AND CPU ID
U 137B. 001D.2000.0180.0800.0010.137C  ;3898  ALU_Q-D,CLK.UBCC  ; CHECK IF THEY ARE THE SAME
U 137C. 0000.013C.0180.0800.0000.116E  ;3899  Z?    ; BRANCH IF YES
U 116E. 0000.003C.0180.0800.0000.1380  ;3900 =0 J/SBE030   ;GO CHECK FOR MX AND TRANSMITTER BIT
U 116F. 0818.0038.0180.0800.0000.137D  ;3901  D_K[.8]    ; ONLY CHECK MX BIT
U 137D. 0B00.003C.01C0.0A70.0000.137E  ;3902  D_D.SWAP,Q_R[0E]
U 137E. 081D.0034.0180.0800.0010.137F  ;3903  D_D.AND.Q,CLK.UBCC  ; CHECK FOR MX BIT CLEAR
U 137F. 0000.013C.0180.0800.0000.1170  ;3904  Z?
U 1170. 0001.003D.0180.09F0.0000.11B9  ;3905 =0 ERROR2,RC[0E]_D   ;DIDN'T SEE MULTILPE XMITTER FAULT CLEAR
        ;3906       ;=======================================
        ;3907       ;DATA: FALUT BITS IN ERROR
        ;3908       ; I.D. UNDER TEST
        ;3909       ;=======================================
U 1171. 0000.003C.0180.0800.0000.1174  ;3910  J/SBE090   ;REPEAT WITH NEXT ID
        ;3911
U 1380. 0000.003C.01C0.0A70.0000.1381  ;3912 SBE030: Q_R[0E]    ; GET FAULT BITS READ
U 1381. 0818.0038.8580.0800.0000.1382  ;3913  D_K[.C]    ; MAASK
U 1382. 0B00.003C.0180.0800.0000.1383  ;3914  D_D.SWAP
U 1383. 001D.0034.01C0.0800.0000.1384  ;3915  Q_D[AND]Q
U 1384. 001D.2000.0180.0800.0010.1385  ;3916  ALU_Q-D,CLK.UBCC  ; CHECK THAT MX AND OTHER BIT IS SET
U 1385. 0000.013C.0180.0800.0000.1172  ;3917  Z?
U 1172. 0001.203D.0180.09F0.0000.11B9  ;3918 =0 ERROR2,RC[0E]_Q
U 1173. 0000.003C.0180.0800.0000.1174  ;3919  NOP
        ;3920 =0
U 1174. 0000.003D.0180.0800.0000.1213  ;3921 SBE090: CALL[CLRFLT]   ; CLEAR THE FAULT REGISTER
U 1175. 0810.0038.0180.0968.0000.1386  ;3922  D_RC[0D]   ;ALL I.D. DONE ???
U 1386. 0019.0020.8D80.0800.0010.1387  ;3923  ALU_D[XOR]K[.1F],CLK.UBCC
U 1387. 0000.013C.0180.0800.0000.1176  ;3924  Z?
U 1176. 0819.0014.0580.09E8.0000.1364  ;3925 =0 RC[0D]_ALU,D_D+K[.1],J/SBE010 ;REPEAT LOOP
U 1177. 0000.003C.0180.0800.0000.1178  ;3926  NOP
        ;3927 ;
        ;3928 ;=============================================================================
        ;3929 ;+
        ;3930 ; SUBTEST A
        ;3931 ;
        ;3932 ; CHECK SBI ERROR CONFIRMATION ON BYTE
        ;3933 ; WRITES TO I/O SPACE.  SEES THAT WRITES TO
        ;3934 ; I/O U-TRAP(STALL).
        ;3935 ;
        ;3936 ; ID=19, "SBI.ERR",BIT #8, CPU SBI ERROR CONFIRMATION
        ;3937 ;
        ;3938 ; ERROR LOGOUT
        ;3939 ;
        ;3940 ; DATA: BYTE DISPLACEMENT BEING USED OFF THE
        ;3941 ;       BASE OF THE  LONG WORD
        ;3942 ;  NOT RELEVENT
        ;3943 ;  DEVICE FLAG (SEE TEST HEADER)
        ;3944 ;  TR LEVEL OF ADAPTER UNDER TEST
        ;3945 ;  TR LEVEL OF MEMORY BEING USED
```

```
            ;3946 ;
            ;3947 ;
            ;3948 ;-
            ;3949 ;================================================================================
            ;3950 ;
            ;3951 =0
U 1178. 0000.003D.0180.0800.0000.11D8  ;3952 M19C03: SUBTEST
U 1179. 0000.003C.0180.0800.0000.117A  ;3953  NOP
U 117A. 0000.003D.0180.0800.0000.11B6  ;3954 =0 ERLOOP
U 117B. 0000.003C.0180.0800.0000.117C  ;3955  NOP
U 117C. 0000.003D.0180.0800.0000.11F8  ;3956 =0 CALL[INIT]
U 117D. 0000.003C.65F0.2C00.0000.1388  ;3957  Q_ID[SBI.ERR]   ;SEE ERR CONFIRMATION CLEAR
U 1388. 0000.003C.0180.0800.0084.7389  ;3958  SC_K[.8]
U 1389. 0001.2024.0180.0800.0010.117E  ;3959  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 117E. 0000.003D.1980.0800.0084.71DA  ;3960 =0 SETRCF[ZERO]
U 117F. 0000.013C.0180.0800.0000.1180  ;3961  Z?
U 1180. 0000.003D.0180.0800.0000.11B7  ;3962 =0 ERROR1    ;ERROR CONFIRMATION SET
            ;3963       ;  DIDN'T CLEAR ON INIT
U 1181. 0003.003C.1180.09F0.0084.7182  ;3964  SC_K[.4],RC[0E]_0
U 1182. 0000.003D.0580.0800.0084.91DA  ;3965 =0 SC_SC+K[.1],CALL[SETRCF]
U 1183. 0000.003C.0180.0800.0000.1184  ;3966  NOP    ;POSITIONS
U 1184. 0000.003D.0180.0800.0000.11B6  ;3967 =0 ERLOOP
U 1185. 0810.0038.0180.0970.0000.138A  ;3968 M19C01: D_RC[0E]
U 138A. 0010.0038.01C0.0900.0000.138B  ;3969  Q_RC[0]
U 138B. 001D.0014.0180.0800.0200.138C  ;3970  VA_ALU,ALU_D[A+B]Q  ;FIND BYTE ADDRESS
U 138C. 0F00.003C.8980.0800.0084.738D  ;3971  D_0,SC_K[.0]   ;CATCH TIMEOUT
U 138D. 0801.001C.0180.0AF8.0000.138E  ;3972  R[0F]_ALU,D_D.ORNOT.MASK
U 138E. 0000.803C.0180.A800.0000.138F  ;3973  MCT/WRITE.P,BYTE
U 138F. 0000.003C.0180.0A78.0010.1390  ;3974  ALU_R[0F],CLK.UBCC  ;DID IT U-TRAP?
U 1390. 0003.013C.0180.0AF8.0000.1186  ;3975  Z?,R[0F]_0
U 1186. 0000.003D.0180.0800.0000.11B7  ;3976 =0 ERROR1    ;DIDN'T U-TRAP OR STALL
U 1187. 0000.003C.65F0.2C00.0000.1391  ;3977  Q_ID[SBI.ERR]   ;CHECK SBI ERROR CONFIRMATION
U 1391. 0000.003C.0180.0800.0084.7392  ;3978  SC_K[.8]   ; BIT SET
U 1392. 0001.2024.0180.0800.0010.1393  ;3979  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1393. 0000.013C.0180.0800.0000.1188  ;3980  Z?
U 1188. 0000.003C.0180.0800.0000.118A  ;3981 =0 J/M19C02   ;BIT WAS SET
U 1189. 0000.003D.0180.0800.0000.11B7  ;3982  ERROR1    ;CPU ERR CONFIRMATION
            ;3983       ; BIT WAS CLEAR
            ;3984 =0
U 118A. 0000.003D.0180.0800.0000.11F8  ;3985 M19C02: CALL[INIT]   ;CLEAR CPU
U 118B. 0000.003C.65F0.2C00.0000.1394  ;3986  Q_ID[SBI.ERR]   ;SEE ERR CONFIRMATION
U 1394. 0000.003C.0180.0800.0084.7395  ;3987  SC_K[.8]   ; CLEAR
U 1395. 0001.2024.0180.0800.0010.1396  ;3988  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 1396. 0000.013C.0180.0800.0000.118C  ;3989  Z?
U 118C. 0000.003D.0180.0800.0000.11B7  ;3990 =0 ERROR1    ;ERR CONFIRMATION DID
            ;3991       ; NOT CLEAR
U 118D. 0810.0038.0180.0970.0000.1397  ;3992  D_RC[0E]   ;CHECK FOR LAST BYTE
U 1397. 0019.0020.0D80.0800.0010.1398  ;3993  ALU_D.XOR.K[.3],CLK.UBCC
U 1398. 0000.013C.0180.0800.0000.118E  ;3994  Z?
U 118E. 0819.0014.0580.09F0.0000.1185  ;3995 =0 RC[0E]_ALU,D_D+K[.1],J/M19C01 ;INCREMENT BYTE DISPLACEMENT
U 118F. 0000.003C.0180.0800.0000.1190  ;3996  NOP    ;END OF TEST
            ;3997 ;
            ;3998 ;================================================================================
            ;3999 ;+
            ;4000 ;
```

```
            ;4001  ; SUBTEST B
            ;4002  ;
            ;4003  ;  SEE THAT EXTENDED WRITES TO I/O SPACE
            ;4004  ;  STALL,U-TRAP AND SET SBI ERROR
            ;4005  ;  COMFIRMATION.
            ;4006  ;
            ;4007  ; ERROR LOGOUT
            ;4008  ;
            ;4009  ; NONE, SEE LISTING FOR FAILING EVENT
            ;4010  ;
            ;4011  ;-
            ;4012  ;=============================================================================
            ;4013  ;
U 1190. 0000.003D.0180.0800.0000.11D8  ;4014 =0 SUBTEST
U 1191. 0000.003C.0180.0800.0000.1192  ;4015  NOP
U 1192. 0000.003D.1980.0800.0084.71DA  ;4016 =0 SETRCF[ZERO]
U 1193. 0000.003C.0180.0800.0000.1194  ;4017  NOP
U 1194. 0000.003D.0180.0800.0000.11B6  ;4018 =0 ERLOOP
U 1195. 0000.003C.0180.0800.0000.1196  ;4019  NOP
U 1196. 0000.003D.0180.0800.0000.11F8  ;4020 =0 CALL[INIT]
U 1197. 0010.0038.0180.0900.0200.1399  ;4021  VA_RC[0]    ;SET I/O ADDRESS
U 1399. 0F00.003C.8980.0800.0084.7198  ;4022  D_0,SC_K[.D]    ;CATCH TIMOUT U-TRAPS
U 1198. 0801.001D.D580.0AF8.0084.71DA  ;4023 =0 R[0F]_ALU,D_D.ORNOT.MASK,SETRCF[.6]
U 1199. 0000.003C.0180.A000.0000.139A  ;4024  MCT/EXTWRITE.P
U 139A. 0000.003C.0180.0A78.0010.139B  ;4025  ALU_R[0F],CLK.UBCC
U 139B. 0000.013C.0180.0800.0000.119A  ;4026  Z?
U 119A. 0000.003D.0180.0800.0000.11B7  ;4027 =0 ERROR1     ;DID NOT U-TRAP
U 119B. 0000.003C.0180.0800.0084.739C  ;4028  SC_K[.8]    ;CHECK ERR CONF IN CPU
U 139C. 0000.003C.65F0.2C00.0000.139D  ;4029  Q_ID[SBI.ERR]    ;READ FAULT
U 139D. 0001.2024.0180.0800.0010.139E  ;4030  ALU_Q.ANDNOT.MASK,CLK.UBCC
U 139E. 0000.013C.0180.0800.0000.119C  ;4031  Z?
U 119C. 0000.003C.0180.0800.0000.139F  ;4032 =0 J/CLN900    ;BIT WAS SET
U 119D. 0000.003D.0180.0800.0000.11B7  ;4033  ERROR1     ;ERROR CONFIRMATION BIT DID NOT SET
            ;4034 CLN900: ;=================================================================
            ;4035  ;*
            ;4036  ; CHECK FOR LAST TR LEVEL (RA 0 = ^X10)
            ;4037  ; ADJUST ADDRESS RC 0,1,2
            ;4038  ;-
            ;4039  ;=================================================================
            ;4040  ;
U 139F. 0810.0038.0180.0958.0000.13A0  ;4041  D_RC[0B]    ;GET CURRENT LEVEL OF TR
U 13A0. 0019.0020.6580.0800.0010.13A1  ;4042  ALU_D.XOR.K[.10],CLK.UBCC ; IS IT = 16
U 13A1. 0000.013C.0180.0800.0000.119E  ;4043  Z?
U 119E. 0000.003C.0180.0800.0000.13A2  ;4044 =0 J/CLN910    ;NO,FIND NEW TR, AND ADDRESSES
U 119F. 0000.003C.0180.0800.0000.13A9  ;4045  J/CLN999    ;YES END TEST
U 13A2. 0819.0014.0580.09D8.0000.13A3  ;4046 CLN910: RC[0B]_ALU,D_D+K[.1]  ;ADJUST TO NEXT LEVEL
U 13A3. 0818.0038.4580.0800.0000.13A4  ;4047  D_K[.8000]
U 13A4. 0200.003C.0180.0800.0000.13A5  ;4048  D_D.RIGHT2    ; GET ADDRESS AJUSTING CONSTANT
U 13A5. 0010.0038.01C0.0900.0000.13A6  ;4049  Q_RC[0]     ;ADJUST RC 0
U 13A6. 081D.0014.0180.0980.0000.13A7  ;4050  RC[0]_ALU,D_D+Q
U 13A7. 0819.0014.1180.0988.0000.13A8  ;4051  RC[1]_ALU,D_D+K[.4]  ;ADJUST RC 1
U 13A8. 0819.0014.1180.0990.0000.10B6  ;4052  RC[2]_ALU,D_D+K[.4],J/CLN000 ;ADJUST RC 2,REPEAT LOOP
U 13A9. 0000.003C.0180.0800.0000.11A0  ;4053 CLN999: NOP     ;END TEST
            ;4054
```

```
ZZ-ESKAR-V22  TEST 218 RESERVED                                                    SEQ 2237
ESKAR58.MCR                        MICRO2  1P(00)    26-JUL-85  17:11:40          Page   9

            ;4055  .PAGE "TEST 218 RESERVED"
            ;4056 =0
U 11A0, 0000,003D,0180,0800,0000,1063  ;4057 T218: NEWTST
U 11A1, 0000,003C,0180,0800,0000,11A2  ;4058  NOP
            ;4059
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 2238
ESKAR58.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40          Page   9

           ;4060  .PAGE "DUMMY NEWTST"
           ;4061 =0
U 11A2, 0000,003D,0180,0800,0000,1063  ;4062 ENDTST: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2239
ESKAR58.MCR                      MICRO2  1P(00)    26-JUL-85  17:11:40          Page  10
:                              Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1911: 1880 1882= 1901= 2736= 2737= 2738= 1881
U 1008   2545= 2546= 2547= 2548= 2549= 2550= 2551= 2552=
U 1010   2743= 2744= 2745= 1904 1902= 1903= 1905 3203=
U 1018   2749= 2750= 2751= 1906 1919= 1920= 1907 3204=
U 1020   3207= 3208= 3209= 3210= 3211= 3212= 3213= 3214=
U 1028   3215= 3216= 3217= 3218= 3219= 3220= 3221= 3222=
U 1030   2755= 2758= 2760= 2761= 2900= 2901= 2902= 2904=
U 1038   2948= 2949= 2950= 2952= 3194= 3195= 3196= 3197=
U 1040   3532= 3533= 3534= 3535= 3539= 3540= 3542= 3543=
U 1048   3609= 3610= 3611= 3612= 1974= 1976= 2015: 190B
U 1050   3655= 3656= 3657= 1909 3708= 3709= 3710= 1910
U 1058   1987= 1988= 2039: 1917 2008= 2009= 2043: 1918
U 1060   3721= 3722= 3724= 1963 3774= 3775= 3776= 3777=
U 1068   2032= 2033= 2233= 2234= 2260= 2261= 2270= 2271=
U 1070   2294= 2295= 2313= 2314= 2320= 2321= 2326= 2327=
U 1078   2337= 2338= 2339= 2341= 2342= 2343= 2344= 2345=
U 1080   2346= 2347= 2348= 2349= 2350= 2351= 2353= 2354=
U 1088   2387= 2388= 2390= 2391= 2406= 2407= 2484= 2485=
U 1090   2524= 2525= 2533= 2534= 2535= 2537= 2540= 2541=
U 1098   2582= 2583= 2584= 2585= 2592= 2593= 2599= 2601=
U 10A0   2604= 2605= 2625= 2626= 2628= 2629= 2795= 2796=
U 10A8   2799= 2800= 2811= 2812= 2815= 2816= 2854= 2855=
U 10B0   3082= 3083= 3084= 3085= 3089= 3091= 3182= 3183=
U 10B8   3184= 3185= 3191= 3192= 3263= 3264= 3265= 3266=
U 10C0   3267= 3268= 3269= 3270= 3271= 3272= 3280= 3281=
U 10C8   3288= 3289= 3290= 3291= 3298= 3299= 3330= 3331=
U 10D0   3332= 3333= 3334= 3335= 3338= 3339= 3345= 3346=
U 10D8   3352= 3353= 3354= 3355= 3357= 3358= 3362= 3363=
U 10E0   3410= 3411= 3412= 3413= 3414= 3415= 3418= 3420=
U 10E8   3428= 3429= 3432= 3433= 3435= 3436= 3441= 3442=
U 10F0   3452= 3453= 3504= 3505= 3506= 3507= 3509= 3510=
U 10F8   3511= 3512= 3513= 3514= 3515= 3516= 3517= 3518=
U 1100   1866: 1867: 1868: 1869: 1870: 1871: 1872: 1873:
U 1108   1874: 1964 3529= 3530= 1875: 1876: 1877: 1878:
U 1110   3537= 3538= 3549= 3550= 3554= 3555= 3586= 3587=
U 1118   3588= 3589= 3591= 3592= 3593= 3594= 3595= 3596=
U 1120   3597= 3598= 3600= 3601= 3602= 3603= 3607= 3608=
U 1128   3617= 3618= 3621= 3622= 3628= 3629= 3630= 3632=
U 1130   3639= 3640= 3641= 3642= 3643= 3644= 3648= 3649=
U 1138   3651= 3652= 3659= 3660= 3688= 3693= 3694= 3695=
U 1140   3697= 3698= 3699= 3700= 3701= 3702= 3703= 3704=
U 1148   3705= 3706= 3717= 3718= 3757= 3758= 3759= 3760=
U 1150   3761= 3762= 3772= 3773= 3781= 3782= 3786= 3787=
U 1158   3795= 3800= 3801= 3802= 3807= 3808= 3811= 3812=
U 1160   3816= 3822= 3851= 3852= 3858= 3860= 3863= 3864=
U 1168   3869= 3870= 3872= 3873= 3877= 3878= 3900= 3901=
U 1170   3905= 3910= 3918= 3919= 3921= 3922= 3925= 3926=
U 1178   3952= 3953= 3954= 3955= 3956= 3957= 3960= 3961=
U 1180   3962= 3964= 3965= 3966= 3967= 3968= 3976= 3977=
U 1188   3981= 3982= 3985= 3986= 3990= 3992= 3995= 3996=
U 1190   4014= 4015= 4016= 4017= 4018= 4019= 4020= 4021=
U 1198   4023= 4024= 4027= 4028= 4032= 4033= 4044= 4045=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  4057= 4058= 4062= 1965 1966 1967 1968 1969
U 11A8  1970 1971 1972 1973 1977 1978 1979 1980
U 11B0  1981 1982 1983 1984 1985 1986 1990 2013
U 11B8  2014 2037 2038 2087 2114 2115 2116 2135
U 11C0  2136 2137 2138 2154 2155 2156 2157 2158
U 11C8  2159 2160 2161 2162 2163 2164 2165 2166
U 11D0  2167 2168 2169 2171 2172 2173 2174 2176
U 11D8  2186 2187 2195 2196 2197 2202 2228 2254
U 11E0  2255 2256 2257 2258 2259 2262 2263 2264
U 11E8  2265 2266 2267 2268 2269 2296 2297 2298
U 11F0  2299 2309 2311 2312 2318 2319 2324 2325
U 11F8  2336 2355 2356 2381 2382 2383 2384 2385
U 1200  2386 2389 2396 2397 2398 2399 2400 2401
U 1208  2402 2403 2404 2405 2416 2417 2418 2426
U 1210  2427 2428 2429 2437 2438 2439 2440 2447
U 1218  2448 2449 2450 2451 2452 2460 2461 2462
U 1220  2463 2470 2471 2472 2473 2480 2481 2482
U 1228  2483 2526 2527 2528 2529 2538 2539 2543
U 1230  2544 2557 2558 2560 2561 2562 2564 2569
U 1238  2570 2571 2573 2578 2580 2581 2589 2590
U 1240  2594 2595 2596 2597 2598 2606 2607 2608
U 1248  2610 2611 2612 2613 2614 2615 2621 2623
U 1250  2624 2627 2652 2653 2654 2655 2684 2685
U 1258  2687 2688 2689 2734 2735 2762 2791 2793
U 1260  2794 2798 2802 2803 2804 2805 2807 2808
U 1268  2809 2810 2813 2814 2843 2844 2845 2846
U 1270  2847 2848 2849 2850 2852 2853 3087 3088
U 1278  3092 3093 3094 3095 3096 3097 3098 3099
U 1280  3100 3101 3102 3103 3104 3105 3106 3107
U 1288  3108 3109 3110 3111 3112 3113 3114 3115
U 1290  3116 3117 3119 3120 3121 3122 3123 3124
U 1298  3125 3127 3128 3129 3130 3131 3133 3134
U 12A0  3135 3136 3138 3139 3140 3141 3142 3143
U 12A8  3145 3146 2198: 3147 3148 3149 3150 3151
U 12B0  3152 3153 3154 3155 3156 3157 3158 3159
U 12B8  3160 3161 3162 3163 3164 3165 3186 3187
U 12C0  3188 3189 3190 3198 3199 3200 3201 3202
U 12C8  3205 3206 3223 3224 3225 3226 3227 3228
U 12D0  3229 3230 3231 3273 3274 3275 3276 3277
U 12D8  3278 3282 3283 3284 3285 3286 3287 3292
U 12E0  3293 3294 3295 3296 3297 3336 3337 3340
U 12E8  3341 3342 3343 3347 3348 3349 3350 3351
U 12F0  3356 3359 3360 3361 3416 3417 3421 3422
U 12F8  3423 3424 3425 3426 3427 3430 3431 3434
U 1300  3437 3438 3439 3440 3443 3444 3445 3446
U 1308  3447 3448 3449 3450 3451 3508 3519 3520
U 1310  3521 3522 3523 3524 3525 3526 3527 3528
U 1318  3546 3547 3548 3590 3604 3605 3613 3614
U 1320  3615 3616 3619 3620 3623 3624 3625 3627
U 1328  3633 3634 3635 3636 3637 3638 3645 3646
U 1330  3647 3650 3653 3654 3658 3696 3707 3711
U 1338  3712 3713 3716 3725 3763 3764 3765 3766
U 1340  3767 3768 3769 3770 3771 3778 3779 3780
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3783 3784 3785 3788 3789 3790 3791 3792
U 1350  3793 3794 3803 3804 3805 3806 3809 3810
U 1358  3813 3814 3815 3853 3854 3855 3856 3861
U 1360  3862 3866 3867 3868 3871 3874 3875 3876
U 1368  3879 3880 3881 3882 3883 3884 3885 3886
U 1370  3887 3888 3889 3890 3891 3892 3893 3894
U 1378  3895 3896 3897 3898 3899 3902 3903 3904
U 1380  3912 3913 3914 3915 3916 3917 3923 3924
U 1388  3958 3959 3969 3970 3971 3972 3973 3974
U 1390  3975 3978 3979 3980 3987 3988 3989 3993
U 1398  3994 4022 4025 4026 4029 4030 4031 4041
U 13A0  4042 4043 4046 4047 4048 4049 4050 4051
U 13A8  4052 4053
U 13B0 - 13EF Unused
U 13F0  2049: 2050: 2051: 2052: 2053: 2054: 2055: 2056:
U 13F8  2057: 2058: 2059: 2060: 2061: 2062: 2063: 2064:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR58	954
Used	954
Remaining	

Total microwords used in memory U: 954
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2243
ESKAR58.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40     Page  10


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR58.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2244
ESKAR59.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 Micro Trap Handler and LSI-11 Support Routines
; 1844   Micro Trap Handler
; 1924 Micro Monitor Interface Routines
; 1925   Newtest Routine
; 1981   Error Loop Routine
; 1998   Error 1 Routine
; 2023   ERROR1 WITH PARAMETER
; 2044   Error 2 Routine
; 2070   ERROR 2 WITH PARAMETER
; 2089   Micro-Monitor Call
; 2103   Reserved Control Store
; 2130   Set Argument Count
; 2153   Increment Subtest Number
; 2174   SBI Unjam Routine
; 2223   RESET SUBTEST NUMBER
; 2245   Initialize the SBI
; 2272   Disable Cache Routine
; 2293   Enable Cache
; 2314   Wait Loop Routine
; 2347   Check for Interrupt Routine
; 2394   CHECK UTRAP
; 2426   CHECK SBI.ERR
; 2456   Set Trap Mask
; 2486   Enable SBI
; 2508   Generate IO Address
; 2534   Select Controller
; 2577 TEST 219 CLEANUP CODE FOR MEMORY TESTS
; 2778 TEST 21A RESERVED
; 2783 TEST 21B RESERVED
; 2788 TEST 21C RESERVED

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR59
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
;1841   .PAGE
;1842
;1843   .TOC " Micro Trap Handler and LSI-11 Support Routines"
;1844   .TOC "    Micro Trap Handler"
;1845 ;**
;1846 ; FUNCTIONAL DESCRIPTION:
;1847 ;
;1848 ; This routine is responsible for 'catching' micro-traps
;1849 ; and reporting unexpected traps to the Monitor. Register R[0F]
;1850 ; contains the Trap-Expected Mask. If the specified bit is set in
;1851 ; the mask, the bit is cleared and a RETURN0 is made otherwise,
;1852 ; an error call is made.
;1853 ;
;1854 ; INPUT PARAMETERS:
;1855 ;
;1856 ; R[0F] - Expected Trap Mask
;1857 ;  Trap Code Function
;1858 ;  ---------      --------
;1859 ;       0  System Init
;1860 ;       1  Data Unaligned
;1861 ;       2  Cross Page
;1862 ;       3  TB Modify
;1863 ;       4  TB Protection
;1864 ;       6  TB Miss
;1865 ;       7  TB Parity
;1866 ;       8  Cache Parity
;1867 ;       9  Reserved Floating Operand
;1868 ;       C  Read Data Substitute
;1869 ;       D  Time out
;1870 ;       F  Control Store Parity Error
;1871 ;      10  Odd Address
;1872 ;
;1873 ; OUTPUT PARAMETERS:
;1874 ;
;1875 ; R[0F] - Mask bit is cleared.
;1876 ;
;1877 ; DATA: Micro PC of Micro Trap
;1878 ;  Trap Code (See Above)
;1879 ;  Fault Status Register
;1880 ;  SBI Error Register
;1881 ;  Timeout Address Register
;1882 ;  Maintenance Register
;1883 ;  Cache Parity Register
;1884 ;  TB Error Register 1
;1885 ;  TB Error Register 0
;1886 ;--
U 1100, 0000,003C,0180,0800,0084,7003  ;1887 1100: sc_k[0],j/trph1  ; System INIT
U 1101, 0000,003C,0580,0800,0084,7001  ;1888 1101: sc_k[.1],j/trph0 ; Data Unaligned
U 1102, 0000,003C,0980,0800,0084,7001  ;1889 1102: sc_k[.2],j/trph0 ; Cross Page
U 1103, 0000,003C,0D80,0800,0084,7001  ;1890 1103: sc_k[.3],j/trph0 ; TB Modify
U 1104, 0000,003C,1180,0800,0084,7001  ;1891 1104: sc_k[.4],j/trph0 ; TB Protection
U 1105, 0000,003C,D580,0800,0084,7001  ;1892 1105: sc_k[.6],j/trph0 ; TB Miss
U 1106, 0000,003C,D980,0800,0084,7001  ;1893 1106: sc_k[.9],j/trph0 ; Reserved Floating Point
U 1107, 0000,003C,5D80,0800,0084,7001  ;1894 1107: sc_k[.7],j/trph0 ; TB Parity Error
U 1108, 0000,003C,0180,0800,0084,7001  ;1895 1108: sc_k[.8],j/trph0 ; Cache Parity Error
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 2249
ESKAR59.MCR                    MICRO2  1P(00)    26-JUL-85  17:12:33                   Page   5

U 110C, 0000.003C.8580.0800.0084.7001  ;1896 110C: sc_k[.c],j/trph0 ; Read Data Substitute
U 110D, 0000.003C.8980.0800.0084.7001  ;1897 110D: sc_k[.d],j/trph0 ; SBI Timeout
U 110E, 0000.003C.6580.0800.0084.7001  ;1898 110E: sc_k[.10],j/trph0 ; Odd Address Trap
U 110F, 0000.003C.6180.0800.0084.7003  ;1899 110F: sc_k[.f],j/trph1 ; Control Store Parity Error
          ;1900
U 1001, 0000.003C.81F0.2C00.0000.1013  ;1901 trph0: q_id[ustack]  ; Get upc of trap
U 1013, 0C00.003C.01E0.0800.0000.1017  ;1902  q_d,d_q   ; Setup to put it back on stack
U 1017, 0C00.003C.81E0.3C00.0000.104F  ;1903  id[ustack]_d,q_d,d_q ; Put upc back onto micro stack
U 104F, 0000.003C.01C0.0A78.0000.1059  ;1904  q_r[0f]   ; Get the Expected Trap Mask
          ;1905  alu_q.andnot.mask,
U 1059, 0001.2024.0180.0800.0010.105B  ;1906   clk.ubcc ; Was trap expected?
U 105B, 0000.013C.0180.0800.0000.1002  ;1907  z?
          ;1908 =0 r[0f]_alu,  ; It was, clear the mask and return
          ;1909       alu_q.and.mask, ; ...
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1910       return0  ; ...
U 1003, 0018.0038.1D80.09E8.0000.101C  ;1911 trph1: rc[0d]_sc  ; Output the Trap Code
U 101C, 0000.003D.D980.0800.0084.707E  ;1912 =0 setrcf[.9]  ; Set output count
U 101D, 0000.003C.49F0.2C00.0000.105C  ;1913  q_id[tber0]  ; Output all the error registers
U 105C, 0001.203C.4DF0.2DB0.0000.105D  ;1914  rc[6]_q,q_id[tber1] ; ...
U 105D, 0001.203C.65F0.2DB8.0000.105F  ;1915  rc[7]_q,q_id[sbi.err] ; ...
U 105F, 0001.203C.6DF0.2DD8.0000.1060  ;1916  rc[0b]_q,q_id[fault] ; ...
U 1060, 0001.203C.75F0.2DE0.0000.1061  ;1917  rc[0c]_q,q_id[maint] ; ...
U 1061, 0001.203C.79F0.2DC8.0000.1062  ;1918  rc[9]_q,q_id[parity] ; ...
U 1062, 0001.203C.69F0.2DC0.0000.1063  ;1919  rc[8]_q,q_id[time.addr] ; ...
U 1063, 0001.203C.81F0.2DD0.0000.1000  ;1920  rc[0a]_q,q_id[ustack] ; ...
          ;1921 1000: ERROR1[.C],  ;
U 1000, 0001.203D.8580.09F0.0084.7022  ;1922   rc[0e]_q ; Report the error
```

```
                ;1923  .PAGE
                ;1924  .TOC " Micro Monitor Interface Routines"
                ;1925  .TOC "   Newtest Routine"
                ;1926 ;**
                ;1927 ; FUNCTIONAL DESCRIPTION:
                ;1928 ;
                ;1929 ; This routine Initializes the following registers:
                ;1930 ;  PSL    to 1F
                ;1931 ;  CES    to 0
                ;1932 ;  ACC.CS    to disable accellerator
                ;1933 ;  SIR    to 0
                ;1934 ;  CLK.CS    to stop interval timer
                ;1935 ;  TBER0    to disable the TB
                ;1936 ;  SBI.MAINT to     0
                ;1937 ;  SBI.ERR    to 0
                ;1938 ;  FAULT    to 0
                ;1939 ;  COMP    to 0
                ;1940 ;  RC[0E]    to  0
                ;1941 ;  R[0F]    to 0
                ;1942 ; It also performs an SBI UNJAM and disables the CACHE.
                ;1943 ; A SCOPE call is then made to the Monitor.
                ;1944 ;
                ;1945 ; CALLING CONSTRAINT:
                ;1946 ;
                ;1947 ; =0
                ;1948 ;
                ;1949 ; SIDE EFFECTS:
                ;1950 ;
                ;1951 ; D Reg is destroyed
                ;1952 ; SC is destroyed
                ;1953 ;--
U 1064, 0000.003C.65F8,0800.0084.7065  ;1954 SCOPE: sc_k[.10],q_0  ; Set IPL to 1F
U 1065, 0818.0038.8D80,0800.0000.1066  ;1955  d_k[.1f]  ; D=1F
U 1066, 0D00.003C.0180,0800.0000.1067  ;1956  d_dal.sc  ; D=001F0000
U 1067, 0000.003C.3D80,3C00.0000.1068  ;1957  id[psl]_d  ; psl = 001F0000
U 1068, 0818.0038.0580,0800.0000.1069  ;1958  d_k[.1]  ; Clear the CES register
U 1069, 0D00.003C.0180,0800.0000.106A  ;1959  d_dal.sc  ; D = 00010000
U 106A, 0F00.003C.3180,3C00.0000.106B  ;1960  id[ces]_d,d_0  ; ces = 00010000
U 106B, 0000.003C.5D80,3C00.0000.106C  ;1961  id[acc.cs]_d  ; acc.cs = 0
U 106C, 0000.003C.3980,3C00.0000.106D  ;1962  id[sir]_d  ; sir = 0
U 106D, 0000.003C.2980,3C00.0000.106E  ;1963  id[clk.cs]_d  ; clk.cs = 0
U 106E, 0000.003C.4980,3C00.0000.101E  ;1964  id[tber0]_d  ; tber0 = 0
U 101E, 0000.003D.0180,0800.0000.1083  ;1965 =0 call[sbi.unjam]  ; Unjam the SBI
                ;1966  intrpt.strobe,  ; Strobe interrupts
U 101F, 0818.0038.C180,0800.4000.106F  ;1967      d_k[.ffff]  ; Setup to clear SBI error register and
U 106F, 0801.0028.6580,3C00.0000.1070  ;1968  id[sbi.err]_d,d_not.d ; and fault register
U 1070, 0F00.003C.6D80,3C00.0000.1071  ;1969  id[fault]_d,d_0  ; ...
U 1071, 0000.003C.6D80,3C00.0000.1072  ;1970  id[fault]_d  ; fault = 0
U 1072, 0000.003C.7580,3C00.0000.1073  ;1971  id[maint]_d  ; MAINT = 0
U 1073, 0000.003C.6580,3C00.0000.1074  ;1972  id[sbi.err]_d  ; sbi error reg = 0
U 1074, 0000.003C.7180,3C00.0000.1075  ;1973  id[comp]_d  ; comp reg = 0
U 1075, 0000.003C.E580,3C00.0000.1076  ;1974  ID[39]_d  ; Clear code for subroutine START.TEST
U 1076, 0001.003C.0180,09F0.0000.1077  ;1975  rc[0e]_d  ;
U 1077, 0001.003C.0180,0AF8.0000.1078  ;1976  r[0f]_d  ; Clear expected trap mask
U 1078, 0001.003C.0180,09F8.0000.1020  ;1977  rc[0f]_d  ; Clear subtest number and argumnet count
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 2251
ESKAR59.MCR                    MICRO2  1P(00)    26-JUL-85  17:12:33                    Page   5

U 1020, 0000,003D,0180,0800,0000,1089  ;1978 =0 call[disable.cache] ; Disable the cache
U 1021, 0F03,003C,0180,09E8,0000,105E  ;1979  rc[0d]_0,d_0,j/callcon ; Call the Micro Monitor
```

```
            :1980   .PAGE
            :1981   .TOC "    Error Loop Routine"
            :1982 ;**
            :1983 ; FUNCTIONAL DESCRIPTION:
            :1984 ;
            :1985 ; This routine is called with the "ERLOOP" macro. The
            :1986 ; routine calls the Micro Monitor to setup an error loop.
            :1987 ;
            :1988 ; CALLING CONSTRAINT:
            :1989 ;
            :1990 ; =0
            :1991 ;
            :1992 ; SIDE EFFECTS:
            :1993 ;
            :1994 ; D Reg - Destroyed
            :1995 ;--
U 1079, 0818,0038,0980,0800,0000,105E   :1996 ERLOOP: d_k[.2],j/callcon
```

```
            ;1997  .PAGE
            ;1998  .TOC "   Error 1 Routine"
            ;1999 ;++
            ;2000 ; FUNCTIONAL DESCRIPTION:
            ;2001 ;
            ;2002 ; This routine is used to report an error to the user. This
            ;2003 ; routine is used when you do not wish to continue in the
            ;2004 ; same test after the call to the Monitor.
            ;2005 ;
            ;2006 ; CALLING CONSTRAINT:
            ;2007 ;
            ;2008 ; None
            ;2009 ;
            ;2010 ; INPUTS:
            ;2011 ;
            ;2012 ; rc[0f]<15:8> - Contain the subtest number
            ;2013 ;
            ;2014 ; OUTPUTS:
            ;2015 ;
            ;2016 ; D<15:8> - Subtest number
            ;2017 ; D<7:0> - Error 1 Monitor Code
            ;2018 ;--
U 107A. 0810,0038,0180,0978,0000,107B  ;2019 ERROR1: d_rc[0f]  ; Get the subtest number
U 107B. 0819,0034,4D80,0800,0000,104E  ;2020  d_d.and.k[.ff00] ; ...
U 104E. 0819,0030,1180,0800,0000,105E  ;2021 104E: d_d.or.k[.4],j/callcon ; Call the monitor
```

```
            ;2022  .PAGE
            ;2023  .TOC "   ERROR1 WITH PARAMETER"
            ;2024 ;++
            ;2025 ; FUNCTIONAL DESCRIPTION:
            ;2026 ;
            ;2027 ; This subroutine takes  as parameter the number of arguments
            ;2028 ; to be printed to the console in the case of an error logout.
            ;2029 ;
            ;2030 ; CALLING CONSTRAINT:
            ;2031 ;
            ;2032 ; =0
            ;2033 ;INPUTS:
            ;2034 ;
            ;2035 ; RC[0F] GETS NUMBER OF PARAMETERS TO BE PRINTED
            ;2036 ;--
            ;2037 =0
            ;2038 ERR1.ARG:
U 1022, 0000.003D.0180.0800.0000.107E  ;2039  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1023, 0000.003C.0180.0800.0000.107A  ;2040  J/ERROR1  ; Go to ERROR1
            ;2041 ;
            ;2042 ;
```

```
            ;2043  .PAGE
            ;2044  .TOC "    Error 2 Routine"
            ;2045 ;++
            ;2046 ; FUNCTIONAL DESCRIPTION:
            ;2047 ;
            ;2048 ; This routine is used  to report an error to the user. This
            ;2049 ; routine is used when you wish to continue in the same test
            ;2050 ;       after the call to the Monitor.
            ;2051 ;
            ;2052 ; CALLING CONSTRAINT:
            ;2053 ;
            ;2054 ; =0
            ;2055 ;
            ;2056 ; INPUTS:
            ;2057 ;
            ;2058 ; rc[0f]<15:8> - Contain the subtest number
            ;2059 ;
            ;2060 ; OUTPUTS:
            ;2061 ;
            ;2062 ; D<15:8> - Subtest number
            ;2063 ; D<7:0> - Error 2 Monitor Code
            ;2064 ;--
U 107C, 0810,0038,0180,0978,0000,107D  ;2065 ERROR2: d_rc[0f]  ; Get the subtest number
U 107D, 0819,0034,4D80,0800,0000,105A  ;2066  d_d.and.k[.ff00] ; ...
U 105A, 0819,0030,D580,0800,0000,105E  ;2067 105A: d_d.or.k[.6],j/callcon ; Call the monitor
            ;2068 ;
```

```
            ;2069  .PAGE
            ;2070  .TOC "   ERROR 2 WITH PARAMETER"
            ;2071 ;**
            ;2072 ; FUNCTIONAL DESCRIPTION:
            ;2073 ;
            ;2074 ; This  is  similar to the  subroutine  ERR1.ARG  defined above,
            ;2075 ; except that in this case after setting the number of arguments
            ;2076 ; too the console, control is transferred to routine ERROR2.
            ;2077 ;
            ;2078 ; INPUTS:
            ;2079 ;
            ;2080 ;       RC[0F] Gets the number of parameters too be printed on the console
            ;2081 ;
            ;2082 ;--
            ;2083 =0
            ;2084 ERR2.ARG:
U 1024, 0000,003D,0180,0800,0000,107E  ;2085  CALL[SETRCF]  ; Set the number of arguments in RC[0F]
U 1025, 0000,003C,0180,0800,0000,107C  ;2086  J/ERROR2  ; Go to ERROR2
            ;2087 ;
```

```
            ;2088  .PAGE
            ;2089  .TOC "   Micro-Monitor Call"
            ;2090 ;*
            ;2091 ; FUNCTIONAL DESCRIPTION:
            ;2092 ;
            ;2093 ; This micro word calls the micro monitor.
            ;2094 ;
            ;2095 ; EXPLICIT INPUTS:
            ;2096 ;
            ;2097 ; D reg must contain valid monitor code.
            ;2098 ;--
            ;2099 105E:
            ;2100 CALLCON:
U 105E, 0000,003C,1D80,3C00,0000,105E  ;2101  id[txdb]_d,j/callcon ; Send code to monitor and hang
```

```
            ;2102  .PAGE
            ;2103  .TOC "   Reserved Control Store"
            ;2104 ;++
            ;2105 ; FUNCTIONAL DESCRIPTION:
            ;2106 ;
            ;2107 ; The following 16 locations must be reserved so the monitor
            ;2108 ; can use them to perform various functions that require
            ;2109 ; the execution of micro-code.
            ;2110 ;--
U 13F0, 0000.003C.0180.0800.0000.13F1  ;2111 13F0: nop
U 13F1, 0000.003C.0180.0800.C000.13F2  ;2112 13F1: nop
U 13F2, 0000.003C.0180.0800.0000.13F3  ;2113 13F2: nop
U 13F3, 0000.003C.0180.0800.0000.13F4  ;2114 13F3: nop
U 13F4, 0000.003C.0180.0800.0000.13F5  ;2115 13F4: nop
U 13F5, 0000.003C.0180.0800.0000.13F6  ;2116 13F5: nop
U 13F6, 0000.003C.0180.0800.0000.13F7  ;2117 13F6: nop
U 13F7, 0000.003C.0180.0800.0000.13F8  ;2118 13F7: nop
U 13F8, 0000.003C.0180.0800.0000.13F9  ;2119 13F8: nop
U 13F9, 0000.003C.0180.0800.0000.13FA  ;2120 13F9: nop
U 13FA, 0000.003C.0180.0800.0000.13FB  ;2121 13FA: nop
U 13FB, 0000.003C.0180.0800.0000.13FC  ;2122 13FB: nop
U 13FC, 0000.003C.0180.0800.0000.13FD  ;2123 13FC: nop
U 13FD, 0000.003C.0180.0800.0000.13FE  ;2124 13FD: nop
U 13FE, 0000.003C.0180.0800.0000.13FF  ;2125 13FE: nop
U 13FF, 0000.003C.0180.0800.0000.12AA  ;2126 13FF: nop
U 12AA, 0000.003C.0180.0800.0000.107E  ;2127 12AA: nop ; This location is permanently blasted in the FPLA
            ;2128   ; to cause a micro ECO to location 12AA.
```

```
            ;2129  .PAGE
            ;2130  .TOC "   Set Argument Count"
            ;2131 ;**
            ;2132 ; FUNCTIONAL DESCRIPTION:
            ;2133 ;
            ;2134 ; This routine is used to set the argument count (RC[0F]<7:0>)
            ;2135 ; to the console.
            ;2136 ;
            ;2137 ; CALLING CONSTRAINT:
            ;2138 ;
            ;2139 ; =0
            ;2140 ;
            ;2141 ; INPUTS:
            ;2142 ;
            ;2143 ; SC - Contains new value of argument count
            ;2144 ;
            ;2145 ; SIDE EFFECTS:
            ;2146 ;
            ;2147 ; Q is destroyed
            ;2148 ;--
U 107E, 0010,0038,01C0,0978,0000,107F  ;2149 SETRCF: q_rc[0f]  ; Get the argument count
U 107F, 0019,2034,4DC0,0800,0000,1080  ;2150  q_q.and.k[.ff00] ; ...
U 1080, 0019,2032,1D80,09F8,0000,0001  ;2151  rc[0f]_q.or.sc,return1 ; Set new argument count and return
```

```
            ;2152  .PAGE
            ;2153  .TOC "   Increment Subtest Number"
            ;2154 ;++
            ;2155 ; FUNCTIONAL DESCRIPTION:
            ;2156 ;
            ;2157 ; This routine is used to increment the subtest number
            ;2158 ; in RC[0F]<15:8>.
            ;2159 ;
            ;2160 ; CALLING CONSTRAINT:
            ;2161 ;
            ;2162 ; =0
            ;2163 ;
            ;2164 ; SIDE EFFECTS:
            ;2165 ;
            ;2166 ; D register is destroyed
            ;2167 ;--
            ;2168 subtst:
U 1081, 0810,0038,4D80,0978,0000,1082  ;2169  d_rc[0f],kmx/.ff00 ; Get current subtest number
            ;2170  rc[0f]_d+k[.ff00], ; Increment and return
U 1082, 0019,4016,4D80,09F8,0000,0001  ;2171      word,return1 ; (Subtest number is negative)
            ;2172
```

```
            ;2173  .PAGE
            ;2174  .TOC "   SBI Unjam Routine"
            ;2175 ;++
            ;2176 ; FUNCTIONAL DESCRIPTION:
            ;2177 ;
            ;2178 ; This routine performs an SBI UNJAM sequence. This is done by
            ;2179 ; asserting HOLD for 16 cycles, then HOLD & UNJAM for 16 cycles
            ;2180 ; and finally HOLD for the last 16 cycles.
            ;2181 ;
            ;2182 ; CALLING CONSTRAINT:
            ;2183 ;
            ;2184 ; =0
            ;2185 ;
            ;2186 ; SIDE EFFECTS:
            ;2187 ;
            ;2188 ; D Reg destroyed
            ;2189 ;--
            ;2190   ;
            ;2191   ; assert hold for 16 cycles
            ;2192   ;
            ;2193 SBI.UNJAM:
            ;2194  mct/sbi.hold,d_k[.10], ; assert hold and set the loop count
U 1083, 0818,0038,6580,8000,0010,1026  ;2195      clk.ubcc  ; set micro cc's for next cycle
            ;2196 =0
            ;2197 SBI.UNJAM.1:
            ;2198  mct/sbi.hold,d_d-k[.1], ; assert hold, decrement loop count and
            ;2199       clk.ubcc,z?, ; ...
U 1026, 0819,0100,0580,8000,0010,1026  ;2200      j/sbi.unjam.1 ; test for completion
            ;2201    ;
            ;2202    ; assert hold and unjam for 16 cycles
            ;2203    ;
            ;2204  mct/sbi.hold+unjam, ; assert hold and unjam, set the loop
U 1027, 0818,0038,6580,8800,0010,1028  ;2205      d_k[.10],clk.ubcc ; set cc's for next cycle
            ;2206 =0
            ;2207 SBI.UNJAM.2:
            ;2208  mct/sbi.hold+unjam, ; loop here for 16 cycles
            ;2209       d_d-k[.1],clk.ubcc, ; of hold and unjam
U 1028, 0819,0100,0580,8800,0010,1028  ;2210      z?,j/sbi.unjam.2 ; ...
            ;2211    ;
            ;2212    ; assert hold for 16 cycles
            ;2213    ;
            ;2214  mct/sbi.hold,d_k[.10], ; assert hold and set loop count
U 1029, 0818,0038,6580,8000,0010,102A  ;2215      clk.ubcc  ; and set cc's for next cycle
            ;2216 =0
            ;2217 SBI.UNJAM.3:
            ;2218  mct/sbi.hold,d_d-k[.1], ; assert hold for 16 cycles
            ;2219       clk.ubcc,z?, ; ...
U 102A, 0819,0100,0580,8000,0010,102A  ;2220      j/sbi.unjam.3 ; ...
U 102B, 0000,003E,0180,0800,0000,0001  ;2221  return1   ; exit
```

```
            ;2222  .PAGE
            ;2223  .TOC "   RESET SUBTEST NUMBER"
            ;2224 ;**
            ;2225 ; FUNCTIONAL DESCRIPTION:
            ;2226 ;
            ;2227 ; Since some  of the tests will have to be restarted when two
            ;2228 ; Ms780-E'S  exist on the SBI, it  will be necessary to reset
            ;2229 ; the subtest counter to zero ( only for thoes tests that have
            ;2230 ; more than one subtest). This subroutine may also be necessary
            ;2231 ; when a test is being loop on.
            ;2232 ;
            ;2233 ; CALLING CONSTRAINT:
            ;2234 ;
            ;2235 ; =0
            ;2236 ; SIDE EFFECTS:
            ;2237 ;
            ;2238 ; D is destroyed
            ;2239 ;
            ;2240 ;--
            ;2241 RESET.SUBTST:
            ;2242  RC[0F]_0,    ; Reset subtest number
U 1084, 0003,003E,0180,09F8,0000,0001  ;2243   RETURN1    ; ...and return
```

```
            ;2244  .PAGE
            ;2245  .TOC "   Initialize the SBI"
            ;2246 ;++
            ;2247 ; FUNCTIONAL DESCRIPTION:
            ;2248 ;
            ;2249 ; This routine performs the following functions:
            ;2250 ;
            ;2251 ;  Executes an SBI Unjam
            ;2252 ;  Clears the SBI Fault Latch
            ;2253 ;  Clears the SBI Error Register
            ;2254 ;
            ;2255 ; CALLING CONSTRAINT:
            ;2256 ;
            ;2257 ; =0
            ;2258 ;
            ;2259 ; SIDE EFFECTS:
            ;2260 ;
            ;2261 ; D & Q Register destroyed
            ;2262 ;--
            ;2263 =0
            ;2264 SBI.INIT:
U 102C, 0000,003D,0180,0800,0000,1083  ;2265  call[sbi.unjam]  ; Unjam the SBI
U 102D, 0818,0038,C180,0800,0000,1085  ;2266  d_k[.ffff]  ; Setup to clear SBI ERROR register
U 1085, 0801,0028,6580,3C00,0000,1086  ;2267  id[sbi.err]_d,d_not.d ; Clear a write/one/to/clear bits
U 1086, 0F00,003C,6D80,3C00,0000,1087  ;2268  id[fault]_d,d_0
U 1087, 0000,003C,6D80,3C00,0000,1088  ;2269  id[fault]_d
U 1088, 0000,003E,6580,3C00,0000,0001  ;2270  id[sbi.err]_d,return1 ; Clear remainder of bits and exit
```

```
            ;2271  .PAGE
            ;2272  .TOC "   Disable Cache Routine"
            ;2273 ;++
            ;2274 ; FUNCTIONAL DESCRIPTION:
            ;2275 ;
            ;2276 ; This routine disables cache hits by setting bits <16:15>
            ;2277 ; in the maintenance register.
            ;2278 ;
            ;2279 ; CALLING SEQUENCE:
            ;2280 ;
            ;2281 ; =0
            ;2282 ;
            ;2283 ; SIDE EFFECTS:
            ;2284 ;
            ;2285 ; D & Q Registers destroyed
            ;2286 ;--
            ;2287 DISABLE.CACHE:
U 1089, 0818.0038.4580.0800.0000.108A  ;2288  d_k[.8000]  ; ... and seed constant
U 108A, 0500.003C.0180.0800.0000.108B  ;2289  d_d.left  ; Generate final constant
U 108B, 0819.0030.4580.0800.0000.108C  ;2290  d_d.or.k[.8000]  ; ... = 00018000
U 108C, 0000.003E.7580.3C00.0000.0001  ;2291  id[maint]_d,return1 ; Disable cache and return
```

```
            ;2292  .PAGE
            ;2293  .TOC "    Enable Cache"
            ;2294 ;++
            ;2295 ; FUNCTIONAL DESCRIPTION:
            ;2296 ;
            ;2297 ; This routine enables cache group 0 by clearing the force
            ;2298 ; miss bit in the maintenance register.
            ;2299 ;
            ;2300 ; CALLING CONSTRAINT:
            ;2301 ;
            ;2302 ; =0
            ;2303 ;
            ;2304 ; SIDE EFFECTS:
            ;2305 ;
            ;2306 ; D, Q AND SC Registers destroyed
            ;2307 ;--
            ;2308 ENABLE.CACHE:
U 108D, 0000.003C.75F0.2C00.0000.108E  ;2309  q_id[maint]  ; Get current maintenance register
U 108E, 0000.003C.6580.0800.0084.708F  ;2310  sc_k[.10]  ; Setup to clear bit 16
U 108F, 0801.2034.0180.0800.0000.1090  ;2311  d_q.and.mask  ; Clear bit 16
U 1090, 0000.003E.7580.3C00.0000.0001  ;2312  id[maint]_d,return1 ; Rewrite register and return
```

```
            ;2313  .PAGE
            ;2314  .TOC "    Wait Loop Routine"
            ;2315 ;++
            ;2316 ; FUNCTIONAL DESCRIPTION:
            ;2317 ;
            ;2318 ; As the name implies,  this subroutine is used to set up a wait
            ;2319 ; loop for a specified  number of cycles.  This may be necessary
            ;2320 ; when the micro-program has to wait for another event to finish.
            ;2321 ; When the subroutine is entered Register D should be holding the
            ;2322 ; desired numbered of cycles.
            ;2323 ;
            ;2324 ; CALLING CONSTRAINT:
            ;2325 ;
            ;2326 ; =0
            ;2327 ;
            ;2328 ; INPUTS:
            ;2329 ;
            ;2330 ; d - Contains number of cycles to wait
            ;2331 ; alu.z condition code must not be set
            ;2332 ;
            ;2333 ; SIDE EFFECTS:
            ;2334 ;
            ;2335 ; d is destroyed
            ;2336 ;--
            ;2337 WAIT.LOOP:
U 1091, 0018,0038,6180,0800,0010,102E  ;2338  alu_k[.F],clk.ubcc ; Make sure alu.z condition
            ;2339      ; ...is not set
            ;2340 =0
            ;2341 W.LOOP:
            ;2342  d_d-k[.1],clk.ubcc,z?, ; Decrement count and check for zero
U 102E, 0819,0100,0580,0800,0010,102E  ;2343       j/W.LOOP
U 102F, 0000,003E,0180,0800,0000,0001  ;2344  return1    ; exit
            ;2345
```

```
            ;2346  .PAGE
            ;2347  .TOC "   Check for Interrupt Routine"
            ;2348 ;++
            ;2349 ; FUNCTIONAL DESCRIPTION:
            ;2350 ;
            ;2351 ; This routine is used to check for any interrupts at level 16 or higher.
            ;2352 ; If an interrupt is active, the following registers are setup:
            ;2353 ;  RC[8] - Gets the VECTOR register
            ;2354 ;  RC[7] - Gets the SBI ERROR register
            ;2355 ;  RC[6] - Gets the FAULT register
            ;2356 ;  RC[5] - Gets 0 if no interrupt otherwise, one
            ;2357 ; and a RETURN2 is made. Otherwise, a return1 is made.
            ;2358 ;
            ;2359 ; CALLING CONSTRAINT:
            ;2360 ;
            ;2361 ; =00
            ;2362 ;
            ;2363 ; SIDE EFFECTS:
            ;2364 ;
            ;2365 ; D & Q are destroyed
            ;2366 ; CRD/RDS and FAULT Interrupt Enables are Cleared
            ;2367 ;--
            ;2368 CHECK_INTERRUPT:
            ;2369    ;
            ;2370    ; First, enable the fault and RDS/CRD interrupts
            ;2371    ;
U 1092. 0818,0038,4580,0800,0000,1093  ;2372  d_k[.8000]  ; Get data to set interrupt enable
            ;2373  id[sbi.err]_d,  ; Set CRD/RDS interrupt enable
U 1093. 0500,003C,6580,3C00,0000,1094  ;2374       d_d.left
U 1094. 0100,003C,0180,0800,0000,1095  ;2375  d_d.left2  ; D = 40000
U 1095. 0000,003C,6D80,3C00,0000,1096  ;2376  id[fault]_d  ; Set fault interrupt enable
            ;2377  intrpt.strobe,d_0, ; Strobe interrupts and setup to clear PSL
U 1096. 0F03,003C,0180,09A8,4000,1097  ;2378       rc[5]_0  ; and clear interrupt occurred flag
U 1097. 0000,003C,3D80,3C00,0000,1098  ;2379  id[psl]_d  ; Clear the PSL
U 1098. 0000,003C,6580,3C00,0000,1099  ;2380  id[sbi.err]_d  ; Clear CRD/RDS interrupt enable
U 1099. 0000,003C,6D80,3C00,0000,109A  ;2381  id[fault]_d  ; Clear FAULT Interrupt Enable
U 109A. 0000,0E3C,0180,0800,0000,1004  ;2382  int?   ; Branch if external interrupt is active
U 1004. 0000,003E,0180,0800,0000,0001  ;2383 =100 return1   ; Exit with no interrupt
U 1005. 0000,003C,0180,0800,0000,1007  ;2384  j/check.int.1  ; Interrupt
U 1006. 0000,003E,0180,0800,0000,0001  ;2385  return1   ; No interrupt
            ;2386 =111
            ;2387 CHECK.INT.1:
U 1007. 0000,003C,35F0,2C00,0000,109B  ;2388  q_id[vector]  ; Get ID Vector Register
U 109B. 0001,203C,65F0,2DC0,0000,109C  ;2389  rc[8]_q,q_id[sbi.err] ; Save VECTOR register and get ERROR reg
U 109C. 0001,203C,6DF0,2DB8,0000,109D  ;2390  rc[7]_q,q_id[fault] ; Save ERROR reg and get FAULT reg
U 109D. 0001,203C,0180,09B0,0000,109E  ;2391  rc[6]_q   ; Save falut reg
U 109E. 0018,003A,0580,09A8,0000,0002  ;2392  rc[5]_k[.1],return2 ; Set error flag and return
```

```
            ;2393  .PAGE
            ;2394  .TOC "   CHECK UTRAP"
            ;2395 ;++
            ;2396 ;FUNCTIONAL DESCRIPTION:
            ;2397 ;
            ;2398 ; This subroutine is used to check wether a Utrap occured.
            ;2399 ; It takes the contents of the Save Utrap flags register
            ;2400 ; R[0E], and compares them to the contrnts of the Utrap
            ;2401 ; register, R[0F]. A RETURN1 or RETURN2 will be performed
            ;2402 ; according to the results of the test  (i.e., contents of
            ;2403 ; these registers are equal or not).
            ;2404 ; CALLING CONSTRAINT:
            ;2405 ;
            ;2406 ; =00
            ;2407 ;
            ;2408 ; INPUTS:
            ;2409 ;
            ;2410 ; R[0E]- Saved Utrap Mask
            ;2411 ; R[0F]- Expected Utrap Mask
            ;2412 ;
            ;2413 ;--
            ;2414 CHECK_UTRAP:
U 109F. 0800.003C.0180.0A70.0000.10A0  ;2415  D_R[0E]    ; Reg D is loaded with the trap mask
U 10A0. 0001.003C.0180.09C0.0000.10A1  ;2416  RC[08]_D   ; Save expected Utrap flag for error logout
U 10A1. 0800.003C.0180.0A78.0000.10A2  ;2417  D_R[0F]    ; Get recieved Utrap flag to D reg
U 10A2. 0001.003C.0180.09B8.0000.10A3  ;2418  RC[07]_D   ; Save in RC[07]
            ;2419  ALU_D.XOR.R[0E], ; Check if any Utraps occured
U 10A3. 000D.0020.0180.0A70.0010.10A4  ;2420          CLK.UBCC
U 10A4. 0003.013C.0180.0AF8.0000.1030  ;2421  Z?,R[0F]_0  ; Branch if no traps
U 1030. 0000.003E.0180.0800.0000.0002  ;2422 =0 RETURN2   ; Utrap occured
U 1031. 0000.003E.0180.0800.0000.0001  ;2423  RETURN1    ; No Utrap  return
            ;2424 ;
```

```
            ;2425  .PAGE
            ;2426  .TOC "   CHECK SBI.ERR"
            ;2427 ;++
            ;2428 ; FUNCTIONAL DESCRIPTION:
            ;2429 ;
            ;2430 ; This subroutine can be used to see whether the SBI.ERR Register
            ;2431 ; has logged an error (i.e, CRD, SBI  or  Time-Out).  The calling
            ;2432 ; routine will have to pass to this subroutine in the SC register
            ;2433 ; the number of the error bit to checked.
            ;2434 ;
            ;2435 ; CALLING CONSTRAINT:
            ;2436 ;
            ;2437 ; =00
            ;2438 ;
            ;2439 ; INPUTS:
            ;2440 ;
            ;2441 ; SC with the number of the error bit to be checked
            ;2442 ;
            ;2443 ; OUTPUTS:
            ;2444 ;
            ;2445 ; RC[08]-Contains the SBI.ERROR Register
            ;2446 ;--
            ;2447 CHECK_SBI_ERR:
U 10A5, 0000,003C,65F0,2C00,0000,10A6  ;2448  Q_ID[SBI.ERR]  ; Read contents of SBI.ERR Register
U 10A6, 0001,203C,0180,09C0,0000,10A7  ;2449  RC[08]_Q  ; Save for error logout
            ;2450  ALU_Q[ANDNOT]MASK, ; See if bit pointed by SC is set in Q
U 10A7, 0001,2024,0180,0800,0010,10A8  ;2451   CLK.UBCC ; ...
U 10A8, 0000,013C,0180,0800,0000,1032  ;2452  Z?   ; Check condition codes
U 1032, 0000,003E,0180,0800,0000,0002  ;2453 =0 RETURN2   ; Bit is set
U 1033, 0000,003E,0180,0800,0000,0001  ;2454  RETURN1   ; Bit is not set
```

```
            ;2455  .PAGE
            ;2456  .TOC "    Set Trap Mask"
            ;2457 ;++
            ;2458 ; FUNCTIONAL DESCRIPTION:
            ;2459 ;
            ;2460 ; This routine is used to set a bit in the Micro Trap Mask
            ;2461 ; R[0F].
            ;2462 ;
            ;2463 ; CALLING CONSTRAINT:
            ;2464 ;
            ;2465 ; =0
            ;2466 ;
            ;2467 ; INPUTS:
            ;2468 ;
            ;2469 ; SC - Contains bit number to set.
            ;2470 ;
            ;2471 ; OUTPUTS:
            ;2472 ;
            ;2473 ; rc[0E] - Contains the current trap mask
            ;2474 ;
            ;2475 ; SIDE EFFECTS:
            ;2476 ;
            ;2477 ; d regsiter is destroyed
            ;2478 ;--
            ;2479 SET_TRAP_MASK:
U 10A9, 0800,003C,0180,0A78,0000,10AA  ;2480  d_r[0f]
U 10AA, 0801,001C,0180,0AF8,0000,10AB  ;2481  r[0f]_d.ornot.mask,d_alu ; Set mask
U 10AB, 0001,003E,0180,0AF0,0000,0001  ;2482  r[0E]_d,return1    ; Save mask and return
            ;2483
            ;2484
```

```
            ;2485  .PAGE
            ;2486  .TOC "   Enable SBI"
            ;2487 ;**
            ;2488 ; FUNCTIONAL DESCRIPTION:
            ;2489 ;
            ;2490 ; This routine is used to ensure the SBI is enabled by clearing
            ;2491 ; bit<12> in the maintenance register.
            ;2492 ;
            ;2493 ; CALLING CONSTRAINT:
            ;2494 ;
            ;2495 ; =0
            ;2496 ;
            ;2497 ; SIDE EFFECTS:
            ;2498 ;
            ;2499 ; D & Q Registers destroyed
            ;2500 ;--
            ;2501 ENABLE.SBI:
U 10AC, 0000,003C,75F0,2C00,0000,10AD  ;2502  q_id[maint]  ; Get current value of maintenance reg
U 10AD, 0000,003C,8580,0800,0084,70AE  ;2503  sc_k[.c]  ; Setup to clear bit<12>
U 10AE, 0801,2034,0180,0800,0000,10AF  ;2504  d_q.and.mask  ; Clear bit<12>
U 10AF, 0000,003E,7580,3C00,0000,0001  ;2505  id[maint]_d,return1 ; and exit
            ;2506
```

```
            ;2507  .PAGE
            ;2508  .TOC "   Generate IO Address"
            ;2509 ;++
            ;2510 ; FUNCTIONAL DESCRIPTION:
            ;2511 ;
            ;2512 ; This routine is used to generate an SBI Configuration Register
            ;2513 ; address from a TR level.
            ;2514 ;
            ;2515 ; CALLING CONSTRAINT:
            ;2516 ;
            ;2517 ; =0
            ;2518 ;
            ;2519 ; INPUTS:
            ;2520 ;
            ;2521 ; D - Contains TR level
            ;2522 ;
            ;2523 ; OUTPUTS:
            ;2524 ;
            ;2525 ; D - Contains SBI IO address
            ;2526 ;--
            ;2527 GENERATE.IO:
U 10B0, 0000,003C,89F8,0800,0084,70B1  ;2528  sc_k[.d],q_0  ; Setup to shift TR level 13 bits
U 10B1, 0D00,003C,8D80,0800,0084,70B2  ;2529  d_del.sc,sc_k[.1f] ; Put TR level in place, setup to
U 10B2, 0000,003C,0980,0800,0084,B0B3  ;2530  sc_sc-k[.2]  ; insert bit 29
U 10B3, 0801,001E,0180,0800,0000,0001  ;2531  d_d.ornot.mask,return1 ; and return
            ;2532
```

```
            ;2533  .PAGE
            ;2534  .TOC "   Select Controller"
            ;2535 ;++
            ;2536 ; FUNCTIONAL DESCRIPTION:
            ;2537 ;
            ;2538 ; This routine is used to put the memory system into the
            ;2539 ; specified configuration with respect to controller select.
            ;2540 ; If a misconfiguration error occurs, a RETURN1 is made otherwise,
            ;2541 ; a RETURN2 is made.
            ;2542 ;
            ;2543 ; CALLING CONSTRAINT:
            ;2544 ;
            ;2545 ; =00
            ;2546 ;
            ;2547 ; INPUTS:
            ;2548 ;
            ;2549 ; d - Contains value to write to bits<2:0> of Register A
            ;2550 ; rc[0e] - Address of Register A
            ;2551 ;
            ;2552 ; SIDE EFFECTS:
            ;2553 ;
            ;2554 ; sc,d,va are destroyed
            ;2555 ;--
            ;2556 SELECT.CNTRLR:
U 10B4. 0010.0038.0180.0970.0284.70B5  ;2557  va_rc[0e],sc_k[.8] ; Set va and setup to get enable bit
U 10B5. 0801.001C.0180.0800.0000.10B6  ;2558  d_d.ornot.mask  ; Insert the enable bit into D reg
U 10B6. 0000.003C.0180.A800.0000.10B7  ;2559  cache.p_d[long]  ; Write the configuration Register
U 10B7. 0818.0038.5D80.0800.0000.10B8  ;2560  d_k[.7]   ; Get Misconfiguration bits
U 10B8. 0000.003C.61F8.0800.0084.70B9  ;2561  sc_k[.F],q_0  ; ...
U 10B9. 0D00.003C.0180.0800.0000.10BA  ;2562  d_dal.sc  ; ... D = x38000
U 10BA. 0000.003C.01E0.0800.0000.10BB  ;2563  q_d   ; Save mask
U 10BB. 0000.003C.0180.C800.0000.10BC  ;2564  d[long]_cache.p  ; Read CNFG A Reg and
            ;2565  alu_q[AND]d,  ; See if expected Misconfiguration bit was set
U 10BC. 001D.2034.0180.0800.0010.10BD  ;2566   clk.ubcc ; ...
U 10BD. 0000.013C.0180.0800.0000.1034  ;2567  z?   ; Branch if no
U 1034. 0000.003E.0180.0800.0000.0001  ;2568 =0 RETURN1   ; Bad configuration
U 1035. 0000.003E.0180.0800.0000.0002  ;2569  RETURN2   ; Configuration ok
            ;2570
            ;2571
            ;2572
U 10BE. 0818.0038.0180.0800.0000.105E  ;2573 DELAY: D_K[.8],J/CALLCON
            ;2574
            ;2575
            ;2576
```

```
            ;2577  .PAGE "TEST 219 CLEANUP CODE FOR MEMORY TESTS"
            ;2578 ;**
            ;2579 ; FUNCTIONAL DESCRIPTION:
            ;2580 ;
            ;2581 ; This routine searches all TR levels of the SBI.
            ;2582 ; If a device is found at the present TR, control
            ;2583 ; is transfered to a routine, specific to each
            ;2584 ; possible device on the SBI, which will try to
            ;2585 ; "clean up" the configuration registers.
            ;2586 ;
            ;2587 ; CALLING CONSTRAINT:
            ;2588 ;
            ;2589 ; None
            ;2590 ;
            ;2591 ; OUTPUTS:
            ;2592 ;
            ;2593 ; None
            ;2594 ;
            ;2595 ; SIDE EFFECTS:
            ;2596 ;
            ;2597 ; None that would affect other code.
            ;2598 ;--
            ;2599 =0
U 1036, 0000,003D,0180,0800,0000,1064  ;2600 CLNP: NEWTST    ;
U 1037, 0018,0038,0580,0A80,0000,10BF  ;2601  r[0]_k[.1]  ; Init TR counter
U 10BF, 0003,003C,0180,0A88,0000,1038  ;2602  r[1]_0   ; Clear "Found One Memory" Flag
            ;2603 =0
            ;2604 TR.LOOP1:
U 1038, 0000,003D,0180,0800,0000,102C  ;2605  call[sbi.init]  ; Initialize the SBI
U 1039, 0000,003C,0180,0800,0000,103A  ;2606  nop   ; ...
U 103A, 0000,003D,0180,0800,0000,10AC  ;2607 =0 call[enable.sbi] ; Enable SBI Cycles
U 103B, 0800,003C,0180,0A00,0000,103C  ;2608  d_r[0]   ; Get ready to generate IO address
U 103C, 0000,003D,0180,0800,0000,10B0  ;2609 =0 call[generate.io] ; Generate address of TR level
U 103D, 0001,003C,0180,09F0,0200,10C0  ;2610  va_d,rc[0E]_d  ; Put address in VA and save in RC[0E]
U 10C0, 0000,003C,8980,0800,0084,703E  ;2611  sc_k[.d]  ; Prepare to enable Time-Out uTrap
U 103E, 0000,003D,0180,0800,0000,10A9  ;2612 =0 call[set_trap_mask] ; Enable timeout micro traps
U 103F, 0000,003C,0180,C800,0000,10C1  ;2613  d[long]_cache.p  ; Read the specified tr level
U 10C1, 0000,003C,0180,0A78,0010,10C2  ;2614  alu_r[0f],clk.ubcc ; Check for no response
U 10C2, 0001,013C,0180,0800,0082,1040  ;2615  z?,sc_d   ; Branch if no adapter here
U 1040, 0000,003C,0180,0800,0000,10C3  ;2616 =0 j/t.check.adpt.code ; Check for a memory
U 1041, 0000,003C,0180,0800,0000,1042  ;2617  j/try.next.tr  ; Try next tr
            ;2618 t.check.adpt.code:
U 10C3, 0000,003C,1D80,0800,1404,70C4  ;2619  state_sc.via.kmx ; Setup to branch on bits <6:4>
U 10C4, 0000,163C,0180,0800,0000,1008  ;2620  state7-4?  ; Branch on the adapter type
U 1008, 0000,003C,0180,0800,0000,10CF  ;2621 =1000 j/t.ms780.a  ; MS780-A
U 1009, 0000,003C,0180,0800,0000,10D5  ;2622  j/t.ms780.c  ; MS780-C
U 100A, 0000,003C,0180,0800,0000,10E2  ;2623  j/uba.or.mba  ; MBA or UBA
U 100B, 0000,003C,0180,0800,0000,1042  ;2624  j/try.next.tr  ; Unknown device
U 100C, 0000,003C,0180,0800,0000,10C8  ;2625  j/ma780   ; MA780
U 100D, 0000,003C,0180,0800,0000,1042  ;2626  j/try.next.tr  ; Unknown device
U 100E, 0000,003C,0180,0800,0000,1010  ;2627  j/ms780.e  ; MS780-E
U 100F, 0000,003C,0180,0800,0000,1010  ;2628  j/ms780.e  ; MS780-E
            ;2629 =0
            ;2630 TRY.NEXT.TR:
U 1042, 0000,003D,0180,0800,0000,10BE  ;2631  delay   ; Reset Mic Mon timer
```

```
U 1043. 0000.003C.0180.0880.0000.10C5  ;2632  la_ra[0]  ; Get TR level
U 10C5. 0818.0014.0580.0A80.0000.10C6  ;2633  r[0]_la+k[.1],d_alu ; Increment TR level
          ;2634  alu_d.xor.k[.f],
U 10C6. 0019.0020.6180.0800.0010.10C7  ;2635      clk.ubcc  ; Check if all done
U 10C7. 0000.013C.0180.0800.0000.1044  ;2636  z?   ; Branch if yes
U 1044. 0000.003C.0180.0800.0000.1038  ;2637 =0 j/tr.loop1  ; Try next TR
U 1045. 0000.003C.0180.0800.0000.10E5  ;2638  j/end.cleanup  ; Tested all TR's
          ;2639
```

```
            ;2640  .PAGE
            ;2641 ;**********************************
            ;2642 ;
            ;2643 ; This portion of code is enabled when a
            ;2644 ; MA780 is found on the SBI
            ;2645 ;
            ;2646 ;
            ;2647 MA780:
U 10C8, 0810.0038.0180.0970.0000.10C9  ;2648  d_rc[0e]  ; Point VA to CR of MA780
U 10C9, 0019.0014.8580.0800.0200.10CA  ;2649  va_d+k[.c]  ; ...
U 10CA, 0818.0038.39F8.0800.0000.10CB  ;2650  d_k[.7ff0],q_0  ; Set starting address of
            ;2651     ; ...MA780 to all 1's
U 10CB, 0000.003C.6580.0800.0084.70CC  ;2652  sc_k[.10]
U 10CC, 0D00.003C.0180.0800.0000.10CD  ;2653  d_dal.sc  ; Put starting address in proper position
U 10CD, 0000.003C.0180.A800.0000.10CE  ;2654  cache.p_d[long]  ; Write the starting address
U 10CE, 0000.003C.0180.0800.0000.1042  ;2655  j/try.next.tr  ; Continue looking at other TR's
            ;2656
```

```
            ;2657  .PAGE
            ;2658 ;*********************************
            ;2659 ;
            ;2660 ; This portion of the code is enabled when
            ;2661 ; a MS780-A was found on the SBI.
            ;2662 ;
            ;2663 ;
            ;2664 t.ms780.a:
U 10CF. 0F00.003C.0180.0800.0000.10D0  ;2665  D_0   ; Prepare to write bit 8 (write enable)
U 10D0. 0819.0010.4980.0800.0000.10D1  ;2666  D_D+K[.FF]+1  ; Generate x100 in D Reg
U 10D1. 0010.0038.0180.0970.0200.10D2  ;2667  va_rc[0e]  ; Point VA to CNFG A Reg
U 10D2. 0000.003C.0180.A800.0000.10D3  ;2668  cache.p_d[long]  ; Write CNFG A Reg
            ;2669
U 10D3. 0000.003C.0180.0A08.0010.10D4  ;2670  alu_r[1],clk.ubcc ; See if a memory was previously found
U 10D4. 0000.013C.0180.0800.0000.1046  ;2671  z?   ; ...
U 1046. 0818.0038.5D80.0800.0000.10DB  ;2672 =0 d_k[.7],j/ms.common ; This is not the first MS780 found
            ;2673      ; ...thus its starting address has to be set
            ;2674      ; ...to all 1's
            ;2675  r[1]_k[.1],d_0,  ; This is the first MS780 found:
U 1047. 0F18.0038.0580.0A88.0000.10DB  ;2676   j/ms.common ; ...set "Found One Memory" Flag
            ;2677      ; ...and go to the common portion of code
            ;2678
            ;2679
            ;2680
            ;2681 ;*********************************
            ;2682 ;
            ;2683 ; This portion of code is executed when
            ;2684 ; a MS780-C was found on the SBI.
            ;2685 ;
            ;2686 ;
            ;2687 t.ms780.c:
U 10D5. 0F00.003C.0180.0800.0000.10D6  ;2688  D_0   ; Prepare to write bit 8 (write enable)
U 10D6. 0819.0010.4980.0800.0000.10D7  ;2689  D_D+K[.FF]+1  ; Generate x100 in D Reg
U 10D7. 0010.0038.0180.0970.0200.10D8  ;2690  va_rc[0e]  ; Point VA to CNFG A Reg
U 10D8. 0000.003C.0180.A800.0000.10D9  ;2691  cache.p_d[long]  ; Write CNFG A Reg
U 10D9. 0000.003C.0180.0A08.0010.10DA  ;2692  alu_r[1],clk.ubcc ; See if a memory was previously found
U 10DA. 0000.013C.0180.0800.0000.1048  ;2693  z?   ; ...
U 1048. 0818.0038.0580.0800.0000.10DB  ;2694 =0 d_k[.1],j/ms.common ; This is not the first MS780 found
            ;2695      ; ...thus its starting address has to be set
            ;2696      ; ...to all 1's
            ;2697  r[1]_k[.1],d_0,  ; This is the first MS780 found:
U 1049. 0F18.0038.0580.0A88.0000.10DB  ;2698   j/ms.common ; ...set "Found One Memory" Flag
            ;2699      ; ...and go to the common portion of code
            ;2700
            ;2701
            ;2702 ;*********************************
            ;2703 ;
            ;2704 ; This segment will be executed when
            ;2705 ; an MS780-E is found on the SBI.
            ;2706 =00
            ;2707 MS780.E:
            ;2708  D_K[.4],  ; Try for Internally Interleave
U 1010. 0818.0039.1180.0800.0000.10B4  ;2709   CALL[SELECT.CNTRLR]
U 1011. 0000.003C.0180.0800.0000.1014  ;2710  J/T.214.TRY.LOW  ; Internal Interleave Misconfiguration
U 1012. 0000.003C.0180.0800.0000.101A  ;2711  J/T.214.CONFIG.OK ; OK. Selected Internal Interleave
```

```
            ;2712 =
            ;2713 =00
            ;2714 T.214.TRY.LOW:
            ;2715  D_K[ZERO],  ; Try for the Lower Controller
U 1014, 0818,0039,1980,0800,0000,10B4  ;2716   CALL[SELECT.CNTRLR]
U 1015, 0000,003C,0180,0800,0000,1018  ;2717  J/T.214.TRY.UPR  ; Lower Misconfigured
U 1016, 0000,003C,0180,0800,0000,101A  ;2718  J/T.214.CONFIG.OK ; OK. Selected the Lower Controller
            ;2719 =
            ;2720 =00
            ;2721 T.214.TRY.UPR:
            ;2722  D_K[.2],  ; Try for the Upper Controller
U 1018, 0818,0039,0980,0800,0000,10B4  ;2723   CALL[SELECT.CNTRLR]
            ;2724  D_K[.1],  ; Could not select any Controllers
U 1019, 0818,0038,0580,0800,0000,10DB  ;2725   J/MS.COMMON ; ...so make the starting address
            ;2726     ; ...of this memory all ones
            ;2727 T.214.CONFIG.OK:
U 101A, 0000,003C,0180,0A08,0010,101B  ;2728  alu_r[1],clk.ubcc ; See if a memory was previously found
U 101B, 0000,013C,0180,0800,0000,104A  ;2729  z?    ; ...
U 104A, 0818,0038,0580,0800,0000,10DB  ;2730 =0 d_k[.1],j/ms.common ; This is not the first MS780 found
            ;2731     ; ...thus its starting address has to be set
            ;2732     ; ...to all 1's
            ;2733  r[1]_k[.1],d_0,  ; This is the first MS780 found:
U 104B, 0F18,0038,0580,0A88,0000,10DB  ;2734   j/ms.common ; ...set "Found One Memory" Flag
            ;2735     ; ...and go to the common portion of code
            ;2736
            ;2737
            ;2738
```

```
            ;2739  .PAGE
            ;2740 ;********************************
            ;2741 ;
            ;2742 ; This portion of code is common to all
            ;2743 ; MS780 memory units.
            ;2744 ;
            ;2745 ;
            ;2746 MS.COMMON:
U 10DB. 0001.003C.0180.0800.0010.10DC  ;2747  alu_d,clk.ubcc  ; See whether the starting address
U 10DC. 0000.013C.0180.0800.0000.104C  ;2748  z?   ; ..should be 0's or 1's
U 104C. 0819.0030.7180.0800.0000.104D  ;2749 =0 d_d.or.k[.fff8]  ; Starting address should be all 1's
U 104D. 0819.0030.0580.0800.0000.10DD  ;2750  d_d.or.k[.1]  ; Set bit 14 (CNFG B Reg)
U 10DD. 0010.0038.8980.0970.0284.70DE  ;2751  va_rc[0e],sc_k[.d] ; Point VA to CNFG B Reg
U 10DE. 0000.003C.0180.0803.0080.D0DF  ;2752  sc_sc+1,va_va+4  ; ...
U 10DF. 0D00.003C.0180.0800.0000.10E0  ;2753  d_dal.sc  ; Shift Starting address in position
U 10E0. 0000.003C.0180.A800.0000.10E1  ;2754  cache.p_d[long]  ; Write CNFG B Reg
U 10E1. 0000.003C.0180.0800.0000.1042  ;2755  j/try.next.tr  ; Go and try next TR level
            ;2756
            ;2757
```

```
            ;2758  .PAGE
            ;2759 ;**********************************
            ;2760 ;
            ;2761 ; When a UBA or a MBA is found on the SBI
            ;2762 ; this portion of code will executed.
            ;2763 ;
            ;2764 ;
            ;2765 UBA.OR.MBA:
U 10E2. 0010.0038.0180.0970.0200.10E3  ;2766  va_rc[0e]  ; Point VA to CR of UBA/MBA
U 10E3. 0818.0038.0580.0803.0000.10E4  ;2767  d_k[.1],va_va+4  ; ...
U 10E4. 0000.003C.0180.A800.0000.1050  ;2768  cache.p_d[long]  ; Init UBA/MBA
            ;2769 =0 d_k[.a0],  ; Wait fo UBA/MBA to
U 1050. 0818.0039.2580.0800.0000.1091  ;2770   call[wait.loop] ; ...finish the Init
U 1051. 0000.003C.0180.0800.0000.1042  ;2771  j/try.next.tr  ; Go and check next TR Level
            ;2772
```

```
            :2773  .PAGE
            :2774 END.CLEANUP:
U 10E5. 0000,003C,0180,0800,0000,1052  :2775  nop   ; Finished Clean-up operation
            :2776
            :2777
```

```
            ;2778  .PAGE "TEST 21A RESERVED"
            ;2779 =0
U 1052, 0000,003D,0180,0800,0000,1064  ;2780 T21A: NEWTST
U 1053, 0000,003C,0180,0800,0000,1054  ;2781  NOP
            ;2782
```

```
ZZ-ESKAR-V22  TEST 21B RESERVED                                                    SEQ 2283
ESKAR59.MCR                    MICRO2  1P(00)    26-JUL-85  17:12:33             Page   8

            ;2783  .PAGE "TEST 21B RESERVED"
            ;2784 =0
U 1054. 0000,003D,0180,0800,0000,1064  ;2785 T21B: NEWTST
U 1055. 0000,003C,0180,0800,0000,1056  ;2786  NOP
            ;2787
```

```
          ;2788  .PAGE "TEST 21C RESERVED"
          ;2789 =0
U 1056. 0000.003D.0180.0800.0000.1064  ;2790 T21C: NEWTST
U 1057. 0000.003C.0180.0800.0000.1058  ;2791  NOP
          ;2792
```

```
            ;2793  .PAGE
            ;2794 =0
U 1058, 0000,003D,0180,0800,0000,1064  ;2795 DUM: NEWTST
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1922: 1901 1910= 1911= 2383= 2384= 2385= 2388=
U 1008  2621= 2622= 2623= 2624= 2625= 2626= 2627= 2628=
U 1010  2709= 2710= 2711= 1902 2716= 2717= 2718= 1903
U 1018  2723= 2725= 2728= 2729= 1912= 1913= 1965= 1967=
U 1020  1978= 1979= 2039= 2040= 2085= 2086= 2200= 2205=
U 1028  2210= 2215= 2220= 2221= 2265= 2266= 2343= 2344=
U 1030  2422= 2423= 2453= 2454= 2568= 2569= 2600= 2601=
U 1038  2605= 2606= 2607= 2608= 2609= 2610= 2612= 2613=
U 1040  2616= 2617= 2631= 2632= 2637= 2638= 2672= 2676=
U 1048  2694= 2698= 2730= 2734= 2749= 2750= 2021: 1904
U 1050  2770= 2771= 2780= 2781= 2785= 2786= 2790= 2791=
U 1058  2795= 1906 2067: 1907 1914 1915 2101: 1916
U 1060  1917 1918 1919 1920 1954 1955 1956 1957
U 1068  1958 1959 1960 1961 1962 1963 1964 1968
U 1070  1969 1970 1971 1972 1973 1974 1975 1976
U 1078  1977 1996 2019 2020 2065 2066 2149 2150
U 1080  2151 2169 2171 2195 2243 2267 2268 2269
U 1088  2270 2288 2289 2290 2291 2309 2310 2311
U 1090  2312 2338 2372 2374 2375 2376 2378 2379
U 1098  2380 2381 2382 2389 2390 2391 2392 2415
U 10A0  2416 2417 2418 2420 2421 2448 2449 2451
U 10A8  2452 2480 2481 2482 2502 2503 2504 2505
U 10B0  2528 2529 2530 2531 2557 2558 2559 2560
U 10B8  2561 2562 2563 2564 2566 2567 2573 2602
U 10C0  2611 2614 2615 2619 2620 2633 2635 2636
U 10C8  2648 2649 2650 2652 2653 2654 2655 2665
U 10D0  2666 2667 2668 2670 2671 2688 2689 2690
U 10D8  2691 2692 2693 2747 2748 2751 2752 2753
U 10E0  2754 2755 2766 2767 2768 2775
U 10E8 - 10FF Unused
U 1100  1887: 1888: 1889: 1890: 1891: 1892: 1893: 1894:
U 1108  1895:    1896: 1897: 1898: 1899:
U 1110 - 12A7 Unused
U 12A8    2127:
U 12B0 - 13EF Unused
U 13F0  2111: 2112: 2113: 2114: 2115: 2116: 2117: 2118:
U 13F8  2119: 2120: 2121: 2122: 2123: 2124: 2125: 2126:
```

ZZ-ESKAR-V22 Line Number Index SEQ 2287
ESKAR59.MCR MICRO2 1P(00) 26-JUL-85 17:12:33 Page 9

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR59	260
Used	260
Remaining	

Total microwords used in memory U: 260
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR59.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2289
ESKAR5A.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2422 TEST 21D TEST FPA STATUS REGISTER ID BIT
; 2460 TEST 21E TEST FPA STATUS REGISTER ENABLE BIT
; 2508 TEST 21F MAINTENANCE TRAP FUNCTION TEST
; 2582 TEST 220 FPA USEQUENCER NAD FIELD TEST
; 2646 TEST 221 ACCELERATER CONTROL TRAP TEST 1
; 2709 TEST 222 ACCELERATER CONTROL TRAP TEST 2
; 2777 TEST 223 UBREAK REGISTER AND SYNC TEST
; 2893 TEST 224 TEST FOR FPA IN IRD STATE
; 2946 TEST 225 TEST FPA/CPU SYNC AND WAIT HAND SHAKE
; 3005 TEST 226 ACCELERATER INSTRUCTION TEST
; 3133 TEST 227 BEN/1 TEST
; 3221 TEST 228 A-FORK TEST
; 3366 TEST 229 B-FORK TEST
; 3552 TEST 22A FPA ID BUS/ACC.BUS DATA PATH TEST
; 3598 TEST 22B FPA BUS A TEST
; 3659 TEST 22C RESERVED
; 3664 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808   ;*****************************************************************************
;1809   ;
;1810   ;
;1811   ; MODULE NAME: ESKAR5A
;1812   ;
;1813   ; CREATION DATE:  SEPTEMBER 1982
;1814   ; AUTHOR:  DAN GRIGORE
;1815   ;
;1816   ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817   ;
;1818   ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819   ;
;1820   ; -  WHAT CHANGE WAS MADE
;1821   ;
;1822   ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823   ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824   ;
;1825   ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826   ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827   ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828   ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829   ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830   ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831   ;    ESKAR3C.
;1832   ;
;1833   ; START OF REVISION HISTORY:
;1834   ;
;1835   ;
;1836   ;
;1837   ;
;1838   ;
;1839   ;*****************************************************************************
;1840
```

```
            ;1841  .PAGE
            ;1842
            ;1843 .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100, 0000,003C,1980,0800,0084,7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000,003C,0580,0800,0084,7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102, 0000,003C,0980,0800,0084,7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103, 0000,003C,0D80,0800,0084,7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104, 0000,003C,1180,0800,0084,7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105, 0000,003C,D580,0800,0084,7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106, 0000,003C,D980,0800,0084,7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107, 0000,003C,5D80,0800,0084,7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108, 0000,003C,0180,0800,0084,7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C, 0000,003C,8580,0800,0084,7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D, 0000,003C,8980,0800,0084,7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E, 0000,003C,6580,0800,0084,7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F, 0000,003C,6180,0800,0084,7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000,003C,81F0,2C00,0000,1006  ;1874 TRPH0: Q_ID[USTACK]
U 1006, 0C00,003C,01E0,0800,0000,100E  ;1875  Q_D,D_Q
U 100E, 0000,003C,8180,3C00,0000,101A  ;1876  ID[USTACK]_D
U 101A, 0C00,003C,01E0,0800,0000,101E  ;1877  Q_D,D_Q
U 101E, 0000,003C,01C0,0A78,0000,102A  ;1878  Q_R[0F]
U 102A, 0001,2024,0180,0800,0010,102E  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102E, 0000,013C,0180,0800,0000,1002  ;1880  Z?     ; BRANCH IF NO
U 1002, 0001,2036,0180,0AF8,0000,0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004, 0000.003D.D980.0800.0084.7163  ;1898 =0 SETRCF[.9]
U 1005, 0000.003C.49F0.2C00.0000.103A  ;1899  Q_ID[TBER0]
U 103A, 0001.203C.4DF0.2DB0.0000.103E  ;1900  RC[6]_Q,Q_ID[TBER1]
U 103E, 0001.203C.65F0.2DB8.0000.104A  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 104A, 0001.203C.6DF0.2DD8.0000.1063  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1063, 0001.203C.75F0.2DE0.0000.1109  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1109, 0001.203C.79F0.2DC8.0000.1147  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1147, 0001.203C.69F0.2DC0.0000.1148  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 1148, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.115D  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 1149, 0000.003C.8580.0800.0084.714A  ;1916 SCOPE: SC_K[.C]
U 114A, 0803.001C.0180.0800.0000.100C  ;1917  ALU_NOT.MASK,D_ALU
U 100C, 0000.003D.7580.3C00.0000.1166  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 100D, 0000.003C.65F8.0800.0084.714B  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 114B, 0818.0038.8D80.0800.0000.114C  ;1920  D_K[.1F]    ; ...
U 114C, 0D00.003C.0180.0800.0000.114D  ;1921  D_DAL.SC
U 114D, 0000.003C.3D80.3C00.0000.114E  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 114E, 0818.0038.0580.0800.0000.114F  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 114F, 0D00.003C.0180.0800.0000.1150  ;1924  D_DAL.SC
U 1150, 0F00.003C.3180.3C00.0000.1151  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1151, 0000.003C.0180.0800.0000.1152  ;1926  NOP
U 1152, 0000.003C.3980.3C00.0000.1153  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1153, 0000.003C.2980.3C00.0000.1154  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1154, 0000.003C.4980.3C00.0000.1156  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1156, 0818.0038.C180.0800.0000.1157  ;1930  D_K[.FFFF]
U 1157, 0000.003C.6580.3C00.0000.1158  ;1931  ID[SBI.ERR]_D
U 1158, 0F00.003C.6D80.3C00.0000.1159  ;1932  ID[FAULT]_D,D_0
U 1159, 0000.003C.6D80.3C00.0000.115A  ;1933  ID[FAULT]_D
U 115A, 0000.003C.6580.3C00.0000.115B  ;1934  ID[SBI.ERR]_D
U 115B, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 115C, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 115D, 0810.0038.0180.0978.0000.115E  ;1937 ERROR1: D_RC[0F]
U 115E, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 115F, 0810.0038.0180.0978.0000.1160  ;1940 ERROR2: D_RC[0F]
U 1160, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155, 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 1161, 0810.0038.4D80.0978.0000.1162  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1162, 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 1163, 0010.0038.01C0.0978.0000.1164  ;1980 SETRCF: Q_RC[0F]
U 1164, 0019.2034.4DC0.0800.0000.1165  ;1981  Q_Q.AND.K[.FF00]
U 1165, 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983 ;
            ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;1986 ;   28: NAD/28  ; NOP
            ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;1990 ;  INITIALIZE ITSELF.
            ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;1992 ;
U 1166, 0818.0038.4580.0800.0000.1167  ;1993 FPINIT: D_K[.8000]
U 1167, 0000.003C.5D80.3C00.0000.1018  ;1994  ID[ACC.CS]_D  ; TURN ON FPA
U 1018, 0818.0039.2D80.0800.0000.116B  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1019, 0000.003C.5980.3C00.0000.1168  ;1996  ID[ACC.2]_D
U 1168, 0000.00FC.0380.0800.0000.101C  ;1997  TRAP.FPA  ; TRAP FPA TO 28.
U 101C, 0018.0039.1980.0800.0200.117B  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 101D, 0F00.003C.0180.0800.0000.1024  ;1999  D_0
U 1024, 0000.003D.0180.0800.0000.116B  ;2000 =0 CALL,J/GENTRA
U 1025, 0000.003C.5980.3C00.0000.1169  ;2001  ID[ACC.2]_D
U 1169, 0000.00FC.0380.0800.0000.1028  ;2002  TRAP.FPA  ; TRAP FPA TO 0
U 1028, 0818.0039.2D80.0800.0000.116B  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 1029, 0000.003C.5980.3C00.0000.116A  ;2004  ID[ACC.2]_D
U 116A, 0000.00FE.0380.0800.0000.0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
            ;2006 ;
            ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2008 ;%0
            ;2009 ;$ 0000.0000, 0000.0000
            ;2010 ;
            ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2014 ;
U 116B, 0000.003C.65F8.0800.0084.716C  ;2015 GENTRA: SC_K[.10],Q_0
U 116C, 0D00.003C.8D80.0800.0084.716D  ;2016  D_DAL.SC,SC_K[.1F]
U 116D, 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2018 ;
            ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2021 ;
            ;2022 =0
U 102C, 0F00.003D.0180.0800.0000.116B  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 102D, 0000.003C.5980.3C00.0000.116E  ;2024  ID[ACC.2]_D
U 116E, 0000.00FC.0380.0800.0000.116F  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 116F, 0000.003C.0180.0800.0000.1170  ;2026  NOP
U 1170, 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
            ;2028 ;*
            ;2029 ; THE FOLLOWING SUB ROUTINE IS USED TO CHECK IF THE "TWINKLE" ECO
            ;2030 ; IS INSTALLED ON THIS MACHINE. THIS IS DONE BY CHECKING THE ECO
            ;2031 ; LEVEL OF THE MACHINE AS RECORDED IN THE SYSTEM ID REGISTER. 'TWINKLED'
            ;2032 ; MACHINES WILL HAVE AN ECO LEVEL OF '70' OR GREATER.
            ;2033 ;
            ;2034 ; THIS ROUTINE MUST BE CALLED WITH A CONSTRAINT OF "=00".
            ;2035 ;
            ;2036 ; THE ROUTINE EXITS WITH THE CONTENTS OF RC[0A] IN THE D REGISTER.
            ;2037 ; A "RETURN1" IS MADE IF THE ECO IS PRESENT OTHERWISE, A RETURN2
            ;2038 ; IS MADE.
            ;2039 ;-
            ;2040 CHECK_TWINKLE:
U 1171, 0000.003C.0DF0.2C00.0000.1172  ;2041  Q_ID[SYS.ID]   ; GET SYSTEM ECO LEVEL
U 1172, 0C00.003C.6180.0800.0000.1173  ;2042  D_Q,K[.F]
U 1173, 001B.0000.6180.0800.0082.1174  ;2043  SC_0-K[.F]   ; SETUP TO SHIFT RIGHT
U 1174, 0D00.003C.0180.0800.0000.1034  ;2044  D_DAL.SC   ; PUT IN BITS<8:0>
U 1034, 0818.0038.81E0.0800.0000.1035  ;2045 =0 Q_D,D_K[.3FF]   ; GENERATE FIELD MASK (= 1FF)
U 1035, 0000.003C.D980.0800.0084.7175  ;2046  SC_K[.9]   ; ...
U 1175, 0801.0034.0180.0800.0000.1176  ;2047  D_D.AND.MASK   ; ...
U 1176, 081D.0034.0180.0800.0000.1177  ;2048  D_D.AND.Q   ; MASK
U 1177, 0018.0038.F9C0.0800.0000.1178  ;2049  Q_K[.7E]   ; GENERATE ECO LEVEL OF A "TWINKLED"
U 1178, 0019.2034.CDC0.0800.0000.1179  ;2050  Q_Q.AND.K[.F0]   ; MACHINE
U 1179, 001D.0000.0180.0800.0010.117A  ;2051  ALU_D-Q,CLK.UBCC  ; IS THIS MACHINE TWINKLED?
U 117A, 0810.1B38.0180.0950.0000.1007  ;2052  ALU.N?,D_RC[0A]   ; BRANCH IF NO
U 1007, 0000.003E.0180.0800.0000.0001  ;2053 =0111 RETURN1    ;
U 100F, 0000.003E.0180.0800.0000.0002  ;2054  RETURN2    ; NO
            ;2055
            ;2056 ;
            ;2057 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2058 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2059 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2060 ;
```

```
U 117B. 2000.003C.0180.0800.0000.117C  ;2061 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 117C. 3000.003C.0180.6000.0000.117D  ;2062  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 117D. 0000.003C.0180.F800.0000.117E  ;2063 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 117E. 0000.003C.0180.F800.0000.117F  ;2064  MCT/ALLOW.IB.READ
U 117F. 0000.003C.0180.F800.0000.1180  ;2065  MCT/ALLOW.IB.READ
U 1180. 0000.003C.0180.F800.0000.1181  ;2066  MCT/ALLOW.IB.READ
U 1181. 0000.003C.0180.F800.0000.1182  ;2067  MCT/ALLOW.IB.READ
U 1182. 1000.003C.0180.F800.0000.1183  ;2068  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1183. 0000.003E.0180.0800.0000.0001  ;2069  RETURN1
U 1184. 3000.003C.0180.0800.0000.117D  ;2070 WAITIB: IBC/START,J/IBW
          ;2071 ;
          ;2072 ; THIS TABLE IS USED TO CATCH THE IB CALL 3 BRANCH WHEN IT IS
          ;2073 ; NECESSARY FOR A CALL 3 TO BE USED IN FPA TESTING.
          ;2074 ;
          ;2075 1200:
          ;2076 IRTAB:
U 1200. 0001.203E.59F0.2DE8.0000.0002  ;2077 1200: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1201. 0001.203E.59F0.2DE8.0000.0002  ;2078 1201: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1202. 0001.203E.59F0.2DE8.0000.0002  ;2079 1202: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1203. 0001.203E.59F0.2DE8.0000.0002  ;2080 1203: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1204. 0001.203E.59F0.2DE8.0000.0002  ;2081 1204: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1205. 0001.203E.59F0.2DE8.0000.0002  ;2082 1205: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1206. 0001.203E.59F0.2DE8.0000.0002  ;2083 1206: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1207. 0001.203E.59F0.2DE8.0000.0002  ;2084 1207: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1208. 0001.203E.59F0.2DE8.0000.0002  ;2085 1208: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1209. 0001.203E.59F0.2DE8.0000.0002  ;2086 1209: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120A. 0001.203E.59F0.2DE8.0000.0002  ;2087 120A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120B. 0001.203E.59F0.2DE8.0000.0002  ;2088 120B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120C. 0001.203E.59F0.2DE8.0000.0002  ;2089 120C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120D. 0001.203E.59F0.2DE8.0000.0002  ;2090 120D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120E. 0001.203E.59F0.2DE8.0000.0002  ;2091 120E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120F. 0001.203E.59F0.2DE8.0000.0002  ;2092 120F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1210. 0001.203E.59F0.2DE8.0000.0002  ;2093 1210: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1211. 0001.203E.59F0.2DE8.0000.0002  ;2094 1211: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1212. 0001.203E.59F0.2DE8.0000.0002  ;2095 1212: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1213. 0001.203E.59F0.2DE8.0000.0002  ;2096 1213: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1214. 0001.203E.59F0.2DE8.0000.0002  ;2097 1214: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1215. 0001.203E.59F0.2DE8.0000.0002  ;2098 1215: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1216. 0001.203E.59F0.2DE8.0000.0002  ;2099 1216: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1217. 0001.203E.59F0.2DE8.0000.0002  ;2100 1217: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1218. 0001.203E.59F0.2DE8.0000.0002  ;2101 1218: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1219. 0001.203E.59F0.2DE8.0000.0002  ;2102 1219: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121A. 0001.203E.59F0.2DE8.0000.0002  ;2103 121A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121B. 0001.203E.59F0.2DE8.0000.0002  ;2104 121B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121C. 0001.203E.59F0.2DE8.0000.0002  ;2105 121C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121D. 0001.203E.59F0.2DE8.0000.0002  ;2106 121D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121E. 0001.203E.59F0.2DE8.0000.0002  ;2107 121E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 121F. 0001.203E.59F0.2DE8.0000.0002  ;2108 121F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1220. 0001.203E.59F0.2DE8.0000.0002  ;2109 1220: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1221. 0001.203E.59F0.2DE8.0000.0002  ;2110 1221: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1222. 0001.203E.59F0.2DE8.0000.0002  ;2111 1222: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1223. 0001.203E.59F0.2DE8.0000.0002  ;2112 1223: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1224. 0001.203E.59F0.2DE8.0000.0002  ;2113 1224: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1225. 0001.203E.59F0.2DE8.0000.0002  ;2114 1225: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1226. 0001.203E.59F0.2DE8.0000.0002  ;2115 1226: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 1227. 0001.203E.59F0.2DE8.0000.0002  ;2116 1227: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1228. 0001.203E.59F0.2DE8.0000.0002  ;2117 1228: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1229. 0001.203E.59F0.2DE8.0000.0002  ;2118 1229: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122A. 0001.203E.59F0.2DE8.0000.0002  ;2119 122A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122B. 0001.203E.59F0.2DE8.0000.0002  ;2120 122B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122C. 0001.203E.59F0.2DE8.0000.0002  ;2121 122C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122D. 0001.203E.59F0.2DE8.0000.0002  ;2122 122D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122E. 0001.203E.59F0.2DE8.0000.0002  ;2123 122E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 122F. 0001.203E.59F0.2DE8.0000.0002  ;2124 122F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1230. 0001.203E.59F0.2DE8.0000.0002  ;2125 1230: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1231. 0001.203E.59F0.2DE8.0000.0002  ;2126 1231: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1232. 0001.203E.59F0.2DE8.0000.0002  ;2127 1232: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1233. 0001.203E.59F0.2DE8.0000.0002  ;2128 1233: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1234. 0001.203E.59F0.2DE8.0000.0002  ;2129 1234: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1235. 0001.203E.59F0.2DE8.0000.0002  ;2130 1235: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1236. 0001.203E.59F0.2DE8.0000.0002  ;2131 1236: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1237. 0001.203E.59F0.2DE8.0000.0002  ;2132 1237: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1238. 0001.203E.59F0.2DE8.0000.0002  ;2133 1238: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1239. 0001.203E.59F0.2DE8.0000.0002  ;2134 1239: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123A. 0001.203E.59F0.2DE8.0000.0002  ;2135 123A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123B. 0001.203E.59F0.2DE8.0000.0002  ;2136 123B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123C. 0001.203E.59F0.2DE8.0000.0002  ;2137 123C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123D. 0001.203E.59F0.2DE8.0000.0002  ;2138 123D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123E. 0001.203E.59F0.2DE8.0000.0002  ;2139 123E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 123F. 0001.203E.59F0.2DE8.0000.0002  ;2140 123F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1240. 0001.203E.59F0.2DE8.0000.0002  ;2141 1240: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1241. 0001.203E.59F0.2DE8.0000.0002  ;2142 1241: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1242. 0001.203E.59F0.2DE8.0000.0002  ;2143 1242: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1243. 0001.203E.59F0.2DE8.0000.0002  ;2144 1243: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1244. 0001.203E.59F0.2DE8.0000.0002  ;2145 1244: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1245. 0001.203E.59F0.2DE8.0000.0002  ;2146 1245: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1246. 0001.203E.59F0.2DE8.0000.0002  ;2147 1246: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1247. 0001.203E.59F0.2DE8.0000.0002  ;2148 1247: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1248. 0001.203E.59F0.2DE8.0000.0002  ;2149 1248: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1249. 0001.203E.59F0.2DE8.0000.0002  ;2150 1249: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124A. 0001.203E.59F0.2DE8.0000.0002  ;2151 124A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124B. 0001.203E.59F0.2DE8.0000.0002  ;2152 124B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124C. 0001.203E.59F0.2DE8.0000.0002  ;2153 124C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124D. 0001.203E.59F0.2DE8.0000.0002  ;2154 124D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124E. 0001.203E.59F0.2DE8.0000.0002  ;2155 124E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 124F. 0001.203E.59F0.2DE8.0000.0002  ;2156 124F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1250. 0001.203E.59F0.2DE8.0000.0002  ;2157 1250: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1251. 0001.203E.59F0.2DE8.0000.0002  ;2158 1251: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1252. 0001.203E.59F0.2DE8.0000.0002  ;2159 1252: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1253. 0001.203E.59F0.2DE8.0000.0002  ;2160 1253: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1254. 0001.203E.59F0.2DE8.0000.0002  ;2161 1254: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1255. 0001.203E.59F0.2DE8.0000.0002  ;2162 1255: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1256. 0001.203E.59F0.2DE8.0000.0002  ;2163 1256: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1257. 0001.203E.59F0.2DE8.0000.0002  ;2164 1257: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1258. 0001.203E.59F0.2DE8.0000.0002  ;2165 1258: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1259. 0001.203E.59F0.2DE8.0000.0002  ;2166 1259: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 125A. 0001.203E.59F0.2DE8.0000.0002  ;2167 125A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 125B. 0001.203E.59F0.2DE8.0000.0002  ;2168 125B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 125C. 0001.203E.59F0.2DE8.0000.0002  ;2169 125C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 125D. 0001.203E.59F0.2DE8.0000.0002  ;2170 125D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 125E. 0001.203E.59F0.2DE8.0000.0002  ;2171 125E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125F. 0001.203E.59F0.2DE8.0000.0002  ;2172 125F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1260. 0001.203E.59F0.2DE8.0000.0002  ;2173 1260: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1261. 0001.203E.59F0.2DE8.0000.0002  ;2174 1261: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1262. 0001.203E.59F0.2DE8.0000.0002  ;2175 1262: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1263. 0001.203E.59F0.2DE8.0000.0002  ;2176 1263: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1264. 0001.203E.59F0.2DE8.0000.0002  ;2177 1264: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1265. 0001.203E.59F0.2DE8.0000.0002  ;2178 1265: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1266. 0001.203E.59F0.2DE8.0000.0002  ;2179 1266: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1267. 0001.203E.59F0.2DE8.0000.0002  ;2180 1267: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1268. 0001.203E.59F0.2DE8.0000.0002  ;2181 1268: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1269. 0001.203E.59F0.2DE8.0000.0002  ;2182 1269: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126A. 0001.203E.59F0.2DE8.0000.0002  ;2183 126A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126B. 0001.203E.59F0.2DE8.0000.0002  ;2184 126B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126C. 0001.203E.59F0.2DE8.0000.0002  ;2185 126C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126D. 0001.203E.59F0.2DE8.0000.0002  ;2186 126D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126E. 0001.203E.59F0.2DE8.0000.0002  ;2187 126E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126F. 0001.203E.59F0.2DE8.0000.0002  ;2188 126F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1270. 0001.203E.59F0.2DE8.0000.0002  ;2189 1270: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1271. 0001.203E.59F0.2DE8.0000.0002  ;2190 1271: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1272. 0001.203E.59F0.2DE8.0000.0002  ;2191 1272: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1273. 0001.203E.59F0.2DE8.0000.0002  ;2192 1273: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1274. 0001.203E.59F0.2DE8.0000.0002  ;2193 1274: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1275. 0001.203E.59F0.2DE8.0000.0002  ;2194 1275: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1276. 0001.203E.59F0.2DE8.0000.0002  ;2195 1276: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1277. 0001.203E.59F0.2DE8.0000.0002  ;2196 1277: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1278. 0001.203E.59F0.2DE8.0000.0002  ;2197 1278: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1279. 0001.203E.59F0.2DE8.0000.0002  ;2198 1279: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127A. 0001.203E.59F0.2DE8.0000.0002  ;2199 127A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127B. 0001.203E.59F0.2DE8.0000.0002  ;2200 127B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127C. 0001.203E.59F0.2DE8.0000.0002  ;2201 127C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127D. 0001.203E.59F0.2DE8.0000.0002  ;2202 127D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127E. 0001.203E.59F0.2DE8.0000.0002  ;2203 127E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127F. 0001.203E.59F0.2DE8.0000.0002  ;2204 127F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1280. 0001.203E.59F0.2DE8.0000.0002  ;2205 1280: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1281. 0001.203E.59F0.2DE8.0000.0002  ;2206 1281: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1282. 0001.203E.59F0.2DE8.0000.0002  ;2207 1282: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1283. 0001.203E.59F0.2DE8.0000.0002  ;2208 1283: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1284. 0001.203E.59F0.2DE8.0000.0002  ;2209 1284: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1285. 0001.203E.59F0.2DE8.0000.0002  ;2210 1285: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1286. 0001.203E.59F0.2DE8.0000.0002  ;2211 1286: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1287. 0001.203E.59F0.2DE8.0000.0002  ;2212 1287: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1288. 0001.203E.59F0.2DE8.0000.0002  ;2213 1288: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1289. 0001.203E.59F0.2DE8.0000.0002  ;2214 1289: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128A. 0001.203E.59F0.2DE8.0000.0002  ;2215 128A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128B. 0001.203E.59F0.2DE8.0000.0002  ;2216 128B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128C. 0001.203E.59F0.2DE8.0000.0002  ;2217 128C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128D. 0001.203E.59F0.2DE8.0000.0002  ;2218 128D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128E. 0001.203E.59F0.2DE8.0000.0002  ;2219 128E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128F. 0001.203E.59F0.2DE8.0000.0002  ;2220 128F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1290. 0001.203E.59F0.2DE8.0000.0002  ;2221 1290: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1291. 0001.203E.59F0.2DE8.0000.0002  ;2222 1291: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1292. 0001.203E.59F0.2DE8.0000.0002  ;2223 1292: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1293. 0001.203E.59F0.2DE8.0000.0002  ;2224 1293: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1294. 0001.203E.59F0.2DE8.0000.0002  ;2225 1294: RC[0D]_Q.Q_ID[ACC.2].RETURN2
```

```
U 1295. 0001.203E.59F0.2DE8.0000.0002  ;2226 1295: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1296. 0001.203E.59F0.2DE8.0000.0002  ;2227 1296: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1297. 0001.203E.59F0.2DE8.0000.0002  ;2228 1297: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1298. 0001.203E.59F0.2DE8.0000.0002  ;2229 1298: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1299. 0001.203E.59F0.2DE8.0000.0002  ;2230 1299: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129A. 0001.203E.59F0.2DE8.0000.0002  ;2231 129A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129B. 0001.203E.59F0.2DE8.0000.0002  ;2232 129B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129C. 0001.203E.59F0.2DE8.0000.0002  ;2233 129C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129D. 0001.203E.59F0.2DE8.0000.0002  ;2234 129D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129E. 0001.203E.59F0.2DE8.0000.0002  ;2235 129E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 129F. 0001.203E.59F0.2DE8.0000.0002  ;2236 129F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A0. 0001.203E.59F0.2DE8.0000.0002  ;2237 12A0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A1. 0001.203E.59F0.2DE8.0000.0002  ;2238 12A1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A2. 0001.203E.59F0.2DE8.0000.0002  ;2239 12A2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A3. 0001.203E.59F0.2DE8.0000.0002  ;2240 12A3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A4. 0001.203E.59F0.2DE8.0000.0002  ;2241 12A4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A5. 0001.203E.59F0.2DE8.0000.0002  ;2242 12A5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A6. 0001.203E.59F0.2DE8.0000.0002  ;2243 12A6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A7. 0001.203E.59F0.2DE8.0000.0002  ;2244 12A7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A8. 0001.203E.59F0.2DE8.0000.0002  ;2245 12A8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12A9. 0001.203E.59F0.2DE8.0000.0002  ;2246 12A9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AB. 0001.203E.59F0.2DE8.0000.0002  ;2247 12AB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AC. 0001.203E.59F0.2DE8.0000.0002  ;2248 12AC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AD. 0001.203E.59F0.2DE8.0000.0002  ;2249 12AD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AE. 0001.203E.59F0.2DE8.0000.0002  ;2250 12AE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AF. 0001.203E.59F0.2DE8.0000.0002  ;2251 12AF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B0. 0001.203E.59F0.2DE8.0000.0002  ;2252 12B0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B1. 0001.203E.59F0.2DE8.0000.0002  ;2253 12B1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B2. 0001.203E.59F0.2DE8.0000.0002  ;2254 12B2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B3. 0001.203E.59F0.2DE8.0000.0002  ;2255 12B3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B4. 0001.203E.59F0.2DE8.0000.0002  ;2256 12B4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B5. 0001.203E.59F0.2DE8.0000.0002  ;2257 12B5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B6. 0001.203E.59F0.2DE8.0000.0002  ;2258 12B6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B7. 0001.203E.59F0.2DE8.0000.0002  ;2259 12B7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B8. 0001.203E.59F0.2DE8.0000.0002  ;2260 12B8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B9. 0001.203E.59F0.2DE8.0000.0002  ;2261 12B9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BA. 0001.203E.59F0.2DE8.0000.0002  ;2262 12BA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BB. 0001.203E.59F0.2DE8.0000.0002  ;2263 12BB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BC. 0001.203E.59F0.2DE8.0000.0002  ;2264 12BC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BD. 0001.203E.59F0.2DE8.0000.0002  ;2265 12BD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BE. 0001.203E.59F0.2DE8.0000.0002  ;2266 12BE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BF. 0001.203E.59F0.2DE8.0000.0002  ;2267 12BF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C0. 0001.203E.59F0.2DE8.0000.0002  ;2268 12C0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C1. 0001.203E.59F0.2DE8.0000.0002  ;2269 12C1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C2. 0001.203E.59F0.2DE8.0000.0002  ;2270 12C2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C3. 0001.203E.59F0.2DE8.0000.0002  ;2271 12C3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C4. 0001.203E.59F0.2DE8.0000.0002  ;2272 12C4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C5. 0001.203E.59F0.2DE8.0000.0002  ;2273 12C5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C6. 0001.203E.59F0.2DE8.0000.0002  ;2274 12C6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C7. 0001.203E.59F0.2DE8.0000.0002  ;2275 12C7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C8. 0001.203E.59F0.2DE8.0000.0002  ;2276 12C8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C9. 0001.203E.59F0.2DE8.0000.0002  ;2277 12C9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CA. 0001.203E.59F0.2DE8.0000.0002  ;2278 12CA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CB. 0001.203E.59F0.2DE8.0000.0002  ;2279 12CB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CC. 0001.203E.59F0.2DE8.0000.0002  ;2280 12CC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 12CD. 0001.203E.59F0.2DE8.0000.0002  ;2281 12CD: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12CE. 0001.203E.59F0.2DE8.0000.0002  ;2282 12CE: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12CF. 0001.203E.59F0.2DE8.0000.0002  ;2283 12CF: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D0. 0001.203E.59F0.2DE8.0000.0002  ;2284 12D0: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D1. 0001.203E.59F0.2DE8.0000.0002  ;2285 12D1: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D2. 0001.203E.59F0.2DE8.0000.0002  ;2286 12D2: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D3. 0001.203E.59F0.2DE8.0000.0002  ;2287 12D3: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D4. 0001.203E.59F0.2DE8.0000.0002  ;2288 12D4: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D5. 0001.203E.59F0.2DE8.0000.0002  ;2289 12D5: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D6. 0001.203E.59F0.2DE8.0000.0002  ;2290 12D6: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D7. 0001.203E.59F0.2DE8.0000.0002  ;2291 12D7: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D8. 0001.203E.59F0.2DE8.0000.0002  ;2292 12D8: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12D9. 0001.203E.59F0.2DE8.0000.0002  ;2293 12D9: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DA. 0001.203E.59F0.2DE8.0000.0002  ;2294 12DA: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DB. 0001.203E.59F0.2DE8.0000.0002  ;2295 12DB: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DC. 0001.203E.59F0.2DE8.0000.0002  ;2296 12DC: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DD. 0001.203E.59F0.2DE8.0000.0002  ;2297 12DD: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DE. 0001.203E.59F0.2DE8.0000.0002  ;2298 12DE: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12DF. 0001.203E.59F0.2DE8.0000.0002  ;2299 12DF: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E0. 0001.203E.59F0.2DE8.0000.0002  ;2300 12E0: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E1. 0001.203E.59F0.2DE8.0000.0002  ;2301 12E1: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E2. 0001.203E.59F0.2DE8.0000.0002  ;2302 12E2: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E3. 0001.203E.59F0.2DE8.0000.0002  ;2303 12E3: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E4. 0001.203E.59F0.2DE8.0000.0002  ;2304 12E4: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E5. 0001.203E.59F0.2DE8.0000.0002  ;2305 12E5: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E6. 0001.203E.59F0.2DE8.0000.0002  ;2306 12E6: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E7. 0001.203E.59F0.2DE8.0000.0002  ;2307 12E7: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E8. 0001.203E.59F0.2DE8.0000.0002  ;2308 12E8: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12E9. 0001.203E.59F0.2DE8.0000.0002  ;2309 12E9: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12EA. 0001.203E.59F0.2DE8.0000.0002  ;2310 12EA: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12EB. 0001.203E.59F0.2DE8.0000.0002  ;2311 12EB: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12EC. 0001.203E.59F0.2DE8.0000.0002  ;2312 12EC: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12ED. 0001.203E.59F0.2DE8.0000.0002  ;2313 12ED: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12EE. 0001.203E.59F0.2DE8.0000.0002  ;2314 12EE: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12EF. 0001.203E.59F0.2DE8.0000.0002  ;2315 12EF: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F0. 0001.203E.59F0.2DE8.0000.0002  ;2316 12F0: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F1. 0001.203E.59F0.2DE8.0000.0002  ;2317 12F1: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F2. 0001.203E.59F0.2DE8.0000.0002  ;2318 12F2: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F3. 0001.203E.59F0.2DE8.0000.0002  ;2319 12F3: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F4. 0001.203E.59F0.2DE8.0000.0002  ;2320 12F4: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F5. 0001.203E.59F0.2DE8.0000.0002  ;2321 12F5: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F6. 0001.203E.59F0.2DE8.0000.0002  ;2322 12F6: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F7. 0001.203E.59F0.2DE8.0000.0002  ;2323 12F7: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F8. 0001.203E.59F0.2DE8.0000.0002  ;2324 12F8: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12F9. 0001.203E.59F0.2DE8.0000.0002  ;2325 12F9: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FA. 0001.203E.59F0.2DE8.0000.0002  ;2326 12FA: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FB. 0001.203E.59F0.2DE8.0000.0002  ;2327 12FB: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FC. 0001.203E.59F0.2DE8.0000.0002  ;2328 12FC: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FD. 0001.203E.59F0.2DE8.0000.0002  ;2329 12FD: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FE. 0001.203E.59F0.2DE8.0000.0002  ;2330 12FE: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12FF. 0001.203E.59F0.2DE8.0000.0002  ;2331 12FF: RC[0D]_Q.Q_ID[ACC.2],RETURN2
            ;2332 ;
            ;2333 ; THIS SUBROUTINE IS USED IN THE FPA SCRATCH PAD TEST.
            ;2334 ; IT SETS UP UNIQUE DATA PATTERNS IN EACH REGISTER SO THAT
            ;2335 ; ADDRESSING TESTS CAN BE PERFORMED. THE DATA GENERATED IS:
```

```
            ;2336 ; R[0]  00000000
            ;2337 ; R[1]  11111111
            ;2338 ; R[2]  22222222
            ;2339 ; ...
            ;2340 ; R[0E]  EEEEEEEE
            ;2341 ; R[0F]  FFFFFFFF
            ;2342 ;
U 1185. 0018.0038.61C0.0800.0000.1186  ;2343 RABTAB: Q_K[.F]
U 1186. 0000.003C.ED80.0800.0084.7187  ;2344 RABP1: SC_K[.1B]
U 1187. 0C00.003C.05F8.0800.0084.9188  ;2345  SC_SC+K[.1],D_Q,Q_0
U 1188. 0D00.003C.0180.0800.0000.1189  ;2346  D_DAL.SC
U 1189. 0F00.003C.11E0.0800.0084.718A  ;2347  Q_D,SC_K[.4],D_0
U 118A. 0D00.003C.0180.0800.0000.118B  ;2348  D_DAL.SC
U 118B. 0D00.003C.0180.0800.0000.118C  ;2349  D_DAL.SC
U 118C. 0D00.003C.0180.0800.0000.118D  ;2350  D_DAL.SC
U 118D. 0D00.003C.0180.0800.0000.118E  ;2351  D_DAL.SC
U 118E. 0D00.003C.0180.0800.0000.118F  ;2352  D_DAL.SC
U 118F. 0D00.003C.0180.0800.0000.1190  ;2353  D_DAL.SC
U 1190. 0D00.003C.0180.0800.0000.1191  ;2354  D_DAL.SC
U 1191. 0D00.003C.0180.0800.0000.1192  ;2355  D_DAL.SC
U 1192. 0019.0034.61C0.0800.0092.1193  ;2356  ALU_D[AND]K[.F],SC_ALU,Q_ALU,CLK.UBCC
U 1193. 0001.013C.0180.08E8.0000.1038  ;2357  Z?,ALU_D,R(SC)_ALU
U 1038. 0019.2000.05C0.0800.0000.1186  ;2358 =0 ALU_Q-K[.1],Q_ALU,J/RABP1
U 1039. 0000.003E.0180.0800.0000.0001  ;2359  RETURN1
            ;2360 ; SHIFT D REGISTTER CONTENTS
            ;2361 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
U 1194. 0000.003C.01F8.0800.0000.1195  ;2362 SHIFTD: Q_0
U 1195. 0500.003C.0180.0800.0000.1196  ;2363  D_D.LEFT
U 1196. 0000.003C.0580.0800.1414.B197  ;2364  EALU_STATE-K[.1],CLK.UBCC
U 1197. 0000.123C.0180.0800.0000.101B  ;2365  EALU.Z?
U 101B. 0000.003C.0180.0800.0000.1194  ;2366 =1011 J/SHIFTD
U 101F. 0000.003E.0180.0800.0000.0001  ;2367  RETURN1
U 1198. 0000.003C.01F8.0800.0000.1199  ;2368 SHIFQD: Q_0
U 1199. 0001.2028.01C0.0800.0000.119A  ;2369  Q_NOT.Q
U 119A. 0500.003C.0280.0800.0000.119B  ;2370  D_D.LEFT.Q  ; FLOATING ZERO PATTERN
U 119B. 0000.003C.0580.0800.1414.B19C  ;2371  EALU_STATE-K[.1],CLK.UBCC
U 119C. 0000.123C.0180.0800.0000.102B  ;2372  EALU.Z?
U 102B. 0000.003C.0180.0800.0000.1198  ;2373 =1011 J/SHIFQD
U 102F. 0000.003E.0180.0800.0000.0001  ;2374  RETURN1
            ;2375 ; THIS ROUTINE LOADS EACH NIBBLE OF THE D REGISTER WITH THE PATTERN HELD BY THE SC
U 119D. 0000.003C.01F8.0800.0000.119E  ;2376 SHIFD4: Q_0
U 119E. 0000.003C.0180.0800.1404.719F  ;2377  STATE_K[.8]  ;
U 119F. 0100.003C.0180.0800.0000.11A0  ;2378 SHIFT: D_D.LEFT2
U 11A0. 0100.003C.0180.0800.0000.11A1  ;2379  D_D.LEFT2
U 11A1. 0811.0030.0180.0948.0000.11A2  ;2380  D_D.OR.RC[9]
U 11A2. 0000.003C.0580.0800.1414.B1A3  ;2381  EALU_STATE-K[.1],CLK.UBCC
U 11A3. 0000.123C.0180.0800.0000.103B  ;2382  EALU.Z?
U 103B. 0000.003C.0180.0800.0000.119F  ;2383 =1011 J/SHIFT
U 103F. 0000.003E.0180.0800.0000.0001  ;2384  RETURN1
U 11A4. 0000.003C.0180.0800.0080.D1A5  ;2385 SC.CHK: SC_SC+1
U 11A5. 0000.003C.6580.0800.0014.B1A6  ;2386  EALU_SC-K[.10],CLK.UBCC
U 11A6. 0000.123C.0180.0800.0000.104B  ;2387  EALU.Z?
U 104B. 0000.003E.0180.0800.0000.0001  ;2388 =1011 RETURN1
U 104F. 0000.003E.0180.0800.0000.0002  ;2389  RETURN2
U 11A7. 0819.0030.4580.0800.0000.11A8  ;2390 TRAP: D_D.OR.K[.8000]  ; TRAP ENABLE BIT
```

```
U 11A8. 0000.003C.6580.0800.1404.703C  ;2391  STATE_K[.10]  ; BIT 16
U 103C. 0000.003D.0180.0800.0000.1194  ;2392 =0 CALL[SHIFTD]
U 103D. 0001.003C.0180.09F0.0000.11A9  ;2393  RC[0E]_D  ; EXPECTED ADDRESS
U 11A9. 0000.003C.5980.3C00.0000.11AA  ;2394  ID[ACC.2]_D  ; WRITE TRAP ADDRESS
U 11AA. 0000.00FC.0380.0800.0000.11AB  ;2395  TRAP.FPA  ; TRAP DIRECTIVE
U 11AB. 0000.003C.59F0.2C00.0000.11AC  ;2396  Q_ID[ACC.2]  ; READ FPA MAINTENANCE REGISTER
U 11AC. 0001.203E.0180.09E8.0000.0001  ;2397  RC[0D]_Q,RETURN1 ; SAVE AND RETURN
U 11AD. 0000.007C.0180.0800.0000.11AE  ;2398 STEP: CPSYNC
U 11AE. 0000.003C.0580.0800.1414.B1AF  ;2399  EALU_STATE-K[.1],CLK.UBCC
U 11AF. 0000.123C.0180.0800.0000.105B  ;2400  EALU.Z?
U 105B. 0000.003C.0180.0800.0000.11AD  ;2401 =1011 J/STEP
U 105F. 0000.003E.0180.0800.0000.0001  ;2402  RETURN1
U 11B0. 0818.0038.4180.0800.0000.11B1  ;2403 TRP.81: D_K[.80]
U 11B1. 0819.0014.0580.0800.0000.11A7  ;2404  D_D+K[.1],J/TRAP
U 11B2. 0818.0038.4180.0800.0000.11B3  ;2405 TRP.8F: D_K[.80]
U 11B3. 0000.003C.6180.0800.0000.11B4  ;2406  KMX/.F
U 11B4. 0819.0014.6180.0800.0000.11A7  ;2407  D_D+K[.F],J/TRAP
            ;2408 ;+
            ;2409 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2410 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE IB
            ;2411 ; IS COMPLETELY FULL OF DATA.
            ;2412 ;
            ;2413 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2414 ;-
U 11B5. 3000.003C.0180.6000.0000.11B6  ;2415 IBLOAD: LOAD.IB,IBC/START
U 11B6. 0000.003C.0180.F800.0000.11B7  ;2416 SYNC02: MCT/ALLOW.IB.READ
U 11B7. 0000.003C.0180.F800.0000.11B8  ;2417  MCT/ALLOW.IB.READ
U 11B8. 0000.003C.0180.F800.0000.11B9  ;2418  MCT/ALLOW.IB.READ
U 11B9. 0000.003C.0180.F800.0000.11BA  ;2419  MCT/ALLOW.IB.READ
U 11BA. 0000.003C.0180.F800.0000.11BB  ;2420  MCT/ALLOW.IB.READ
U 11BB. 1000.003E.0180.F800.0000.0001  ;2421  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
```

```
                ;2422  .PAGE "TEST 21D TEST FPA STATUS REGISTER ID BIT"
                ;2423 ;*****************************************************************
                ;2424 ;**
                ;2425 ; TEST ID BIT IN FPA STATUS REGISTER
                ;2426 ;
                ;2427 ; TEST DESCRIPTION
                ;2428 ; THE FPA STATUS REGISTER ID BIT IS TESTED BY READING
                ;2429 ; THE STATUS REGISTER AND MASKING TO PASS THE
                ;2430 ; ID BIT ONLY.
                ;2431 ;
                ;2432 ; LOGIC DESCRIPTION
                ;2433 ; THE FPA ID BUS INTERFACE, AND THE STATUS REGISTER
                ;2434 ; LOGIC ARE ON THE FML AND FMH MODULES.
                ;2435 ;
                ;2436 ; ERROR DESCRIPTION
                ;2437 ; EXPECTED ID BIT
                ;2438 ; RECEIVED ID BIT
                ;2439 ;
                ;2440 ; SYNC POINT DESCRIPTION
                ;2441 ;--
                ;2442 ;*****************************************************************
                ;2443 ;/////////////////////////////////////////
                ;2444 ;*
                ;2445 ; ID BIT TEST
                ;2446 ;-
                ;2447 =0
U 1044, 0018.0039.0980.09F8.0000.1149  ;2448 BTST1: NEWTST[.2]
U 1045, 0000.003C.0180.0800.0000.1048  ;2449  NOP
U 1048, 0000.003D.0180.0800.0000.1166  ;2450 =0 CALL[FPINIT]
U 1049, 0000.003C.5DF0.2C00.0000.11BC  ;2451  Q_ID[ACC.CS] ; READ FPA STATUS REGISTER
U 11BC, 0001.203C.0180.09E8.0000.11BD  ;2452  RC[0D]_Q  ; SAVE
U 11BD, 0818.0038.0580.0800.0000.11BE  ;2453  D_K[.1]  ; MASK FOR ID BIT
U 11BE, 0001.003C.0180.09F0.0000.11BF  ;2454  RC[0E]_D
U 11BF, 001D.0034.0180.0800.0010.11C0  ;2455  ALU_D.AND.Q,CLK.UBCC ; CHECK ID BIT
U 11C0, 0000.013C.0180.0800.0000.104C  ;2456  Z?
U 104C, 0000.003C.0180.0800.0000.11C1  ;2457 =0 J/ENTST1
U 104D, 0000.003D.0180.0800.0000.115D  ;2458  ERROR1   ; ID BIT NOT SET
U 11C1, 0000.003C.0180.0800.0000.1054  ;2459 ENTST1: NOP
```

```
            ;2460  .PAGE "TEST 21E TEST FPA STATUS REGISTER ENABLE BIT"
            ;2461 ;*****************************************************************
            ;2462 ;**
            ;2463 ; TEST FPA STATUS REGISTER ENABLE BIT
            ;2464 ;
            ;2465 ; TEST DESCRIPTION
            ;2466 ; THE FPA STATUS REGISTER ENABLE BIT IS TESTED
            ;2467 ; BY WRITING IT TO A ZERO/ONE AND VERIFYING IT
            ;2468 ; EACH TIME.
            ;2469 ;
            ;2470 ; LOGIC DESCRIPTION
            ;2471 ; THE FPA ID BUS INTERFACE, AND THE STATUS REGISTER
            ;2472 ; LOGIC ARE ON THE FML AND FMH MODULES.
            ;2473 ;
            ;2474 ; ERROR DESCRIPTION
            ;2475 ; EXPECTED ENABLE BIT
            ;2476 ; RECEIVED ENABLE BIT
            ;2477 ; LOOP COUNTER
            ;2478 ;
            ;2479 ; SYNC POINT DESCRIPTION
            ;2480 ;--
            ;2481 ;*****************************************************************
            ;2482 ;/////////////////////////////////////////
            ;2483 ;*
            ;2484 ; SET/RESET ENABLE BIT
            ;2485 ;-
            ;2486 =0
U 1054, 0018,0039,0D80,09F8,0000,1149  ;2487 BTST2: NEWTST[.3]
U 1055, 0000,003C,0180,0800,0000,1058  ;2488  NOP
U 1058, 0000,003D,0180,0800,0000,1166  ;2489 =0 CALL[FPINIT]
U 1059, 0018,0038,0D80,09F8,0000,11C2  ;2490  RC[0F]_K[.3]  ; DISPLAY 3 PARAMETERS
U 11C2, 0003,003C,0180,09F0,0000,11C3  ;2491  RC[0E]_0
U 11C3, 0018,0038,0980,09E0,0000,105C  ;2492  RC[0C]_K[.2]
U 105C, 0000,003D,0180,0800,0000,115C  ;2493 =0 ERLOOP
U 105D, 0810,0038,0180,0970,0000,11C4  ;2494 LOOP: D_RC[0E]
U 11C4, 0000,003C,5D80,3C00,0000,11C5  ;2495  ID[ACC.CS]_D  ; RESET FPA STATUS WORD
U 11C5, 0000,003C,5DF0,2C00,0000,11C6  ;2496  Q_ID[ACC.CS]  ; READ FPA STATUS WORD
U 11C6, 0019,2034,45C0,0800,0000,11C7  ;2497  Q_Q.AND.K[.8000] ; MASK FOR ENABLE BIT
U 11C7, 0001,203C,0180,09E8,0000,11C8  ;2498  RC[0D]_Q  ; SAVE
U 11C8, 001D,0020,0180,0800,0010,11C9  ;2499  ALU_D.XOR.Q,CLK.UBCC ; CHECK ENABLE BIT RST/SET
U 11C9, 0000,013C,0180,0800,0000,1064  ;2500  Z?
U 1064, 0000,003D,0180,0800,0000,115D  ;2501 =0 ERROR1
U 1065, 0810,0038,0180,0960,0000,11CA  ;2502  D_RC[0C]  ; RESTORE D REGISTER
U 11CA, 0018,0038,4580,09F0,0000,11CB  ;2503  RC[0E]_K[.8000]  ; BIT 15
U 11CB, 0019,8000,0580,09E0,0010,11CC  ;2504  RC[0C]_D-K[.1],CLK.UBCC,BYTE
U 11CC, 0000,013C,0180,0800,0000,1066  ;2505  Z?
U 1066, 0000,003C,0180,0800,0000,105D  ;2506 =0 J/LOOP
U 1067, 0000,003C,0180,0800,0000,1068  ;2507 ENTST1A: NOP
```

```
                ;2508  .PAGE "TEST 21F MAINTENANCE TRAP FUNCTION TEST"
                ;2509 ;*****************************************************************
                ;2510 ;+*
                ;2511 ; MAINTENANCE TRAP FUNCTION TEST
                ;2512 ;
                ;2513 ; TEST DESCRIPTION
                ;2514 ;  THIS IS A TEST OF THE TRAP FUNCTION FOR THE FPA USEQUENCER
                ;2515 ;  WHICH IS ACCESSIBLE THROUGH THE FPA MAINTENANCE REGISTER
                ;2516 ;  ON THE ID BUS. THIS ALLOWS THE FPA USEQUENCER TO BE TRAPPED
                ;2517 ;  TO ANY LOCATION IN THE LOWER HALF OF THE FPA ROM (NAD<07:00>
                ;2518 ;  IS USER SPECIFIED AND NAD<08>=0). A 1 IS FLOATED THROUGH 0'S
                ;2519 ;  AND A ZERO IS FLOATED THROUGH 1'S TO GENERATE THE TRAP ADDRESSES
                ;2520 ;  USED IN THIS TEST. TO CHECK THE RESULT OF EACH TRAP FUNCTION
                ;2521 ;  ISSUED THE LOWER (BITS<08:00>) BITS OF THE FPA MAINTENANCE
                ;2522 ;  REGISTER ARE USED.
                ;2523 ;
                ;2524 ; FPA UCODE USED
                ;2525 ;  NONE
                ;2526 ;
                ;2527 ; LOGIC DESCRIPTION
                ;2528 ;
                ;2529 ; ERROR DESCRIPTION
                ;2530 ;  EXPECTED FPA MAINTENANCE REGISTER
                ;2531 ;  GOT FPA MAINTENANCE REGISTER
                ;2532 ;
                ;2533 ; SYNC POINT DESCRIPTION
                ;2534 ;
                ;2535 ;--
                ;2536 ;*****************************************************************
                ;2537 =0
U 1068. 0000.003D.0180.0800.0000.1149  ;2538 TRP1: NEWTST
U 1069. 0018.0038.0980.09F8.0000.11CD  ;2539  RC[0F]_K[.2]
U 11CD. 0018.0038.5D80.0A80.0000.106A  ;2540  R[0]_K[.7]  ; SET UP A COUNTER
                ;2541      ; USED TO GENERATED TEST ADDRESSES.
U 106A. 0000.003D.0180.0800.0000.115C  ;2542 =0 ERLOOP
U 106B. 0000.003C.0180.0800.0000.106C  ;2543  NOP
                ;2544 =0
U 106C. 0000.003D.0180.0800.0000.1166  ;2545 TRP1A: CALL,J/FPINIT  ; GO INITIALIZE THE FPA.
U 106D. 0000.003C.0180.0A00.0082.11CE  ;2546  SC_R[0]   ; GENERATE 1 IN 0'S TEST ADDRESS
U 11CE. 0803.001C.0180.0800.0000.1070  ;2547  D_NOT.MASK
U 1070. 0819.0035.4980.0A88.0000.116B  ;2548 =0 ALU_D.AND.K[.FF],R[1]_ALU,D_ALU,CALL,J/GENTRA ; GO GENERATE
                ;2549      ; A PATTERN TO BE LOADED INTO
                ;2550      ; THE FPA MAINTENANCE REGISTER.
U 1071. 0000.003C.5980.3C00.0000.11CF  ;2551  ID[ACC.2]_D  ; LOAD FPA MAINTENANCE REG.
U 11CF. 0000.00FC.5BF0.2E08.0000.11D0  ;2552  TRAP.FPA,Q_ID[ACC.2],LAB_R[1] ; TRAP HERE AND READ MAINTENANCE REG.
U 11D0. 080D.0030.8D80.0800.0084.71D1  ;2553  ALU_D[OR]LB,D_ALU,SC_K[.1F]
U 11D1. 0801.0034.6180.09F0.0084.71D2  ;2554  ALU_D.AND.MASK,RC[0E]_ALU,D_ALU,SC_K[.F] ; GENERATE EXPECTED FPA MAINT.
U 11D2. 0000.003C.0580.0800.0084.B1D3  ;2555  SC_SC-K[.1]
U 11D3. 0001.2034.01C0.09E8.0000.11D4  ;2556  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU ; SAVE ACTUAL RESULT.
U 11D4. 001D.0000.0180.0800.0010.11D5  ;2557  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 11D5. 0000.013C.0180.0800.0000.1072  ;2558  Z?
U 1072. 0000.003D.0180.0800.0000.115D  ;2559 =0 ERROR1   ; TRAP FUNCTION FAILED OR FPA
                ;2560      ; MAINTENANCE REGISTER BROKEN.
U 1073. 0000.003C.0180.0800.0000.1074  ;2561  NOP
U 1074. 0000.003D.0180.0800.0000.1166  ;2562 =0 CALL,J/FPINIT  ; INITIALIZE FPA
```

```
U 1075. 0000.003C.0180.0A00.0082.11D6  ;2563  SC_R[0]   ; NOW REPEAT TEST WITH 0 IN 1'S ADDRESS
U 11D6. 0800.0038.0180.0800.0000.1076  ;2564  D_MASK
U 1076. 0819.0035.4980.0A88.0000.116B  ;2565 =0 ALU_D.AND.K[.FF],R[1]_ALU,D_ALU,CALL,J/GENTRA ; GO GENERATE
          ;2566     ; PATTERN TO BE LOADED INTO
          ;2567     ; THE FPA MAINTENANCE REG.
U 1077. 0000.003C.5980.3C00.0000.11D7  ;2568  ID[ACC.2]_D  ; LOAD MAINT. REG.
U 11D7. 0000.00FC.5BF0.2E08.0000.11D8  ;2569  TRAP.FPA,Q_ID[ACC.2],LAB_R[1] ; TRAP HERE AND READ MAINT. REG.
U 11D8. 080D.0030.8D80.0800.0084.71D9  ;2570  ALU_D[OR]LB,D_ALU,SC_K[.1F]
U 11D9. 0801.0034.6180.09F0.0084.71DA  ;2571  ALU_D.AND.MASK,RC[0E]_ALU,D_ALU,SC_K[.F] ; GENERATE EXPECTED MAINT. REG.
U 11DA. 0000.003C.0580.0800.0084.B1DB  ;2572  SC_SC-K[.1]
U 11DB. 0001.2034.01C0.09E8.0000.11DC  ;2573  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU ; SAVE ACTUAL RESULT
U 11DC. 001D.0000.0180.0800.0010.11DD  ;2574  ALU_D-Q,CLK.UBCC ; CHECK RESULT
U 11DD. 0000.013C.0180.0800.0000.1078  ;2575  Z?
U 1078. 0000.003D.0180.0800.0000.115D  ;2576 =0 ERROR1   ; UTRAP FUNCTION FAILED OR
          ;2577     ; FPA MAINTENANCE REGISTER BROKEN.
U 1079. 0000.003C.0180.0A00.0010.11DE  ;2578  ALU_R[0],CLK.UBCC ; SEE IF ALL PATTERNS HAVE BEEN TRIED.
U 11DE. 0000.013C.0180.0800.0000.107A  ;2579  Z?
U 107A. 0018.0000.0580.0A80.0000.106C  ;2580 =0 R[0]_LA-K[.1],J/TRP1A ; CONTINUE TEST.
U 107B. 0000.003C.0180.0800.0000.107C  ;2581  NOP
```

```
                ;2582  .PAGE "TEST 220 FPA USEQUENCER NAD FIELD TEST"
                ;2583 ;*****************************************************************
                ;2584 ;**
                ;2585 ; FPA USEQUENCER NAD FIELD TEST
                ;2586 ;
                ;2587 ; TEST DESCRIPTION
                ;2588 ;  THIS IS A TEST OF THE FPA UCODE NAD FIELD.
                ;2589 ;  A 1 IS FLOATED ACROSS 0'S IN THIS FIELD
                ;2590 ;  AND THE MAINTENANCE REGISTER USED TO CHECK
                ;2591 ;  THE RESULTING NEXT ADDRESS.
                ;2592 ;
                ;2593 ; FPA UCODE USED
                ;2594 ;  LOCATION CONTENTS
                ;2595 ;  8:  NAD/100
                ;2596 ;  9:  NAD/80
                ;2597 ;  0A:  NAD/40
                ;2598 ;  0B:  NAD/20
                ;2599 ;  0C:  NAD/10
                ;2600 ;  0D:  NAD/8
                ;2601 ;  0E:  NAD/4
                ;2602 ;  0F:  NAD/2
                ;2603 ;  10:  NAD/1
                ;2604 ;  11:  NAD/0
                ;2605 ;
                ;2606 ; LOGIC DESCRIPTION
                ;2607 ;
                ;2608 ; ERROR DESCRIPTION
                ;2609 ;  EXPECTED FPA MAINTENANCE REG.
                ;2610 ;  GOT FPA MAINTENANCE REG.
                ;2611 ;
                ;2612 ; SYNC POINT DESCRIPTION
                ;2613 ;
                ;2614 ;--
                ;2615 ;*****************************************************************
                ;2616 =0
U 107C. 0000.003D.0180.0800.0000.1149  ;2617 NAD1: NEWTST
U 107D. 0018.0038.0980.09F8.0000.11DF  ;2618  RC[0F]_K[.2]
U 11DF. 0018.0038.0180.0A80.0000.11E0  ;2619  R[0]_K[.8]  ; SET UP A COUNTER.
U 11E0. 0018.0038.0180.0A88.0000.107E  ;2620  R[1]_K[.8]  ; SET UP FIRST TRAP ADDRESS
                ;2621      ; USED TO GET TO THE FPA UCODE.
U 107E. 0000.003D.0180.0800.0000.115C  ;2622 =0 ERLOOP
U 107F. 0000.003C.0180.0800.0000.1080  ;2623  NOP
                ;2624 =0
U 1080. 0000.003D.0180.0800.0000.1166  ;2625 NAD1A: CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1081. 0800.003C.0180.0A08.0000.1082  ;2626  D_R[1]
U 1082. 0000.003D.0180.0800.0000.116B  ;2627 =0 CALL,J/GENTRA
U 1083. 0000.003C.5980.3E00.0082.11E1  ;2628  ID[ACC.2]_D,SC_R[0] ; SET UP TRAP ADDRESS.
U 11E1. 0801.00DC.0380.0800.0000.11E2  ;2629  TRAP,FPA,ALU_D.ORNOT.MASK,D_ALU ; TRAP FPA AND
                ;2630      ; GENERATE EXPECTED NAD SEEN IN
                ;2631      ; FPA MAINTENANCE REGISTER.
U 11E2. 0000.003C.59F0.2C00.0000.11E3  ;2632  Q_ID[ACC.2]  ; GET MAINT. REG.
U 11E3. 0000.003C.8D80.0800.0084.71E4  ;2633  SC_K[.1F]
U 11E4. 0801.0034.6180.09F0.0084.71E5  ;2634  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU,SC_K[.F]
U 11E5. 0000.003C.0580.0800.0084.B1E6  ;2635  SC_SC-K[.1]
U 11E6. 0001.2034.01C0.09E8.0000.11E7  ;2636  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU
```

```
ZZ-ESKAR-V22  TEST 220 FPA USEQUENCER NAD FIELD TEST                                    SEQ 2309
ESKAR5A.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:12                   Page   6

U 11E7. 001D.0000.0180.0800.0010.11E8  :2637  ALU_D-Q,CLK.UBCC ; CORRECT?
U 11E8. 0000.013C.0180.0800.0000.1084  :2638  Z?
U 1084. 0000.003D.0180.0800.0000.115D  :2639 =0 ERROR1    ; NAD FIELD FAILED.
U 1085. 0000.003C.0180.0A08.0000.11E9  :2640  LAB_R[1]   ; INCREMENT UCODE ADDRESS.
U 11E9. 0018.0014.0580.0A88.0000.11EA  :2641  R[1]_LA+K[.1]
U 11EA. 0000.003C.0180.0A00.0010.11EB  :2642  ALU_R[0],CLK.UBCC ; SEE IF TEST DONE.
U 11EB. 0000.013C.0180.0800.0000.1086  :2643  Z?
U 1086. 0018.0000.0580.0A80.0000.1080  :2644 =0 R[0]_LA-K[.1],J/NAD1A ; CONTINUE TEST
U 1087. 0000.003C.0180.0800.0000.1088  :2645  NOP
```

```
            ;2646  .PAGE "TEST 221 ACCELERATER CONTROL TRAP TEST 1"
            ;2647 ;*****************************************************************
            ;2648 ;**
            ;2649 ; ACCELERATER CONTROL TRAP TEST 1
            ;2650 ;
            ;2651 ; TEST DESCRIPTION
            ;2652 ;  THIS IS A TEST OF THE CPU UWORD ACCELERATER CONTROL
            ;2653 ;  FIELD WHICH HAS THE FUNCTION OF TRAPPING
            ;2654 ;  THE FPA TO ONE OF ITS LOCATIONS 0 THRU 7.
            ;2655 ;  THIS TEST TRIES TO USE THIS FUNCTION TO TRAP THE
            ;2656 ;  FPA TO EACH OF LOCATIONS 0 THRU 7. NOTE THAT A TABLE IS
            ;2657 ;  EMPLOYED BY THIS TEST TO VARY THE CPU ACM FIELD.
            ;2658 ;
            ;2659 ; FPA UCODE USED
            ;2660 ;  NONE
            ;2661 ;
            ;2662 ; LOGIC DESCRIPTION
            ;2663 ;
            ;2664 ; ERROR DESCRIPTION
            ;2665 ;  EXPECTED FPA MAINTENANCE REGISTER
            ;2666 ;  GOT FPA MAINTENANCE REGISTER
            ;2667 ;
            ;2668 ; SYNC POINT DESCRIPTION
            ;2669 ;
            ;2670 ;--
            ;2671 ;*****************************************************************
            ;2672 =0
U 1088, 0000,003D,0180,0800,0000,1149  ;2673 TRP2: NEWTST
U 1089, 0018,0038,0980,09F8,0000,11EC  ;2674  RC[0F]_K[.2]
U 11EC, 0018,0038,5D80,09F0,0000,108A  ;2675  RC[0E]_K[.7]  ; INITIALIZE A COUNTER AND
            ;2676      ; THE EXPECTED TRAP ADDRESS.
U 108A, 0000,003D,0180,0800,0000,115C  ;2677 =0 ERLOOP
U 108B, 0000,003C,0180,0800,0000,108C  ;2678  NOP
            ;2679 =0
U 108C, 0000,003D,0180,0800,0000,1166  ;2680 TRP2A: CALL,J/FPINIT
U 108D, 0000,003C,8D80,0800,0084,71ED  ;2681  SC_K[.1F]
U 11ED, 0803,001C,0180,0800,0000,11EE  ;2682  ALU_NOT.MASK,D_ALU
U 11EE, 0000,003C,5980,3C00,0000,11EF  ;2683  ID[ACC.2]_D  ; CLEAR THE FPA MAINT. REG.
U 11EF, 0810,0038,0180,0970,0000,108E  ;2684  D_RC[0E]
U 108E, 0000,193D,0180,0800,0000,1010  ;2685 =0 CALL,D3-0?,J/TRP2TB ; INDEX INTO TABLE WITH ACF AND ACM FUNCTIONS.
U 108F, 0000,003C,6180,0800,0084,71F0  ;2686  SC_K[.F]
U 11F0, 0000,003C,0580,0800,0084,B1F1  ;2687  SC_SC-K[.1]
U 11F1, 0001,2034,01C0,09E8,0000,11F2  ;2688  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU
U 11F2, 001D,0000,0180,0800,0010,11F3  ;2689  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 11F3, 0000,013C,0180,0800,0000,1090  ;2690  Z?
U 1090, 0000,003D,0180,0800,0000,115D  ;2691 =0 ERROR1   ; TRAP FUNCTION OR ADDRESS INCORRECT.
U 1091, 0810,0038,0180,0970,0010,11F4  ;2692  D_RC[0E],CLK.UBCC
U 11F4, 0000,013C,0180,0800,0000,1092  ;2693  Z?
U 1092, 0019,0000,0580,09F0,0000,108C  ;2694 =0 ALU_D-K[.1],RC[0E]_ALU,J/TRP2A ; IF NOT DONE CONTINUE TEST.
U 1093, 0000,003C,0180,0800,0000,11F5  ;2695  J/TRP2B
            ;2696 ; THIS TABLE IS USED TO EXECUTE ACF/TRAP WITH VARYING ACM
            ;2697 ; FIELDS OF 0 THRU 7.
            ;2698 =000
U 1010, 0000,00BE,5870,2C00,0000,0001  ;2699 TRP2TB: ACF/TRAP,ACM/0,Q_ID[ACC.2],RETURN1 ; TRAP TO 0
U 1011, 0000,00BE,58F0,2C00,0000,0001  ;2700  ACF/TRAP,ACM/1,Q_ID[ACC.2],RETURN1 ; TRAP TO 1
```

```
U 1012, 0000,00BE,5970,2C00,0000,0001  ;2701  ACF/TRAP,ACM/2,Q_ID[ACC.2],RETURN1 ; TRAP TO 2
U 1013, 0000,00BE,59F0,2C00,0000,0001  ;2702  ACF/TRAP,ACM/3,Q_ID[ACC.2],RETURN1 ; TRAP TO 3
U 1014, 0000,00BE,5A70,2C00,0000,0001  ;2703  ACF/TRAP,ACM/4,Q_ID[ACC.2],RETURN1 ; TRAP TO 4
U 1015, 0000,00BE,5AF0,2C00,0000,0001  ;2704  ACF/TRAP,ACM/5,Q_ID[ACC.2],RETURN1 ; TRAP TO 5
U 1016, 0000,00BE,5B70,2C00,0000,0001  ;2705  ACF/TRAP,ACM/6,Q_ID[ACC.2],RETURN1 ; TRAP TO 6
U 1017, 0000,00BE,5BF0,2C00,0000,0001  ;2706  ACF/TRAP,ACM/7,Q_ID[ACC.2],RETURN1 ; TRAP TO 7
          ;2707 ; COME HERE WHEN DONE.
U 11F5, 0000,003C,0180,0800,0000,1094  ;2708 TRP2B: NOP
```

```
                ;2709   .PAGE "TEST 222 ACCELERATER CONTROL TRAP TEST 2"
                ;2710  ;*****************************************************************
                ;2711  ;**
                ;2712  ; ACCELERATER CONTROL TRAP TEST 2
                ;2713  ;
                ;2714  ; TEST DESCRIPTION
                ;2715  ;  THIS TEST INSURES THAT NO FPA TRAP IS GENERATED
                ;2716  ;  BY CONDITIONS IN THE CPU ACF FIELD WHICH SHOULD
                ;2717  ;  NOT CAUSE A TRAP. NOTE THAT A TABLE IS USED TO EXECUTE
                ;2718  ;  ACF FIELDS OF DIFFERENT VALUES EACH OF WHICH
                ;2719  ;  SHOULD NOT CAUSE A TRAP.
                ;2720  ;
                ;2721  ; FPA UCODE USED
                ;2722  ;  LOCATION CONTENTS
                ;2723  ;  28:  NAD/28  ; NOP
                ;2724  ;
                ;2725  ; LOGIC DESCRIPTION
                ;2726  ;
                ;2727  ; ERROR DESCRIPTION
                ;2728  ;  EXPECTED FPA MAINTENANCE REGISTER
                ;2729  ;  GOT FPA MAINTENANCE REGISTER
                ;2730  ;  COUNTER IDENTIFYING WHICH ACF AND ACM FIELD VALUE FAILED:
                ;2731  ;   COUNT ACF ACM
                ;2732  ;    3 3 6
                ;2733  ;    2 3 5
                ;2734  ;    1 3 3
                ;2735  ;    0 1 7
                ;2736  ;
                ;2737  ; SYNC POINT DESCRIPTION
                ;2738  ;
                ;2739  ;--
                ;2740  ;*****************************************************************
                ;2741  =0
U 1094. 0000.003D.0180.0800.0000.1149   ;2742 TRP3: NEWTST
U 1095. 0018.0038.0D80.09F8.0000.11F6   ;2743  RC[0F]_K[.3]
U 11F6. 0018.0038.0D80.09E0.0000.1096   ;2744  RC[0C]_K[.3]  ; COUNTER
U 1096. 0000.003D.0180.0800.0000.115C   ;2745 =0 ERLOOP
U 1097. 0000.003C.0180.0800.0000.1098   ;2746  NOP
                ;2747 =0
U 1098. 0000.003D.0180.0800.0000.1166   ;2748 TRP3A: CALL,J/FPINIT  ; SHOULD LEAVE THE FPA AT 28
U 1099. 0818.0038.4980.0800.0000.109A   ;2749  D_K[.FF]
U 109A. 0000.003D.0180.0800.0000.116B   ;2750 =0 CALL,J/GENTRA
U 109B. 0000.003C.5980.3C00.0000.11F7   ;2751  ID[ACC.2]_D
U 11F7. 0819.0030.2D80.0800.0000.11F8   ;2752  ALU_D.OR.K[.28],D_ALU
U 11F8. 0000.003C.8D80.0800.0084.71F9   ;2753  SC_K[.1F]
U 11F9. 0801.0034.0180.09F0.0000.11FA   ;2754  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU ; COMPUTE EXPECTED MAINT. REG.
U 11FA. 0810.0038.0180.0960.0000.109C   ;2755  D_RC[0C]  ; GO EXECUTE THE
U 109C. 0000.193D.0180.0800.0000.1008   ;2756 =0 CALL,D3-0?,J/TRP3TB ; INSTRUCTION CONTAINING THE
                ;2757      ; ACF AND ACM FUNCTION BEING TESTED.
U 109D. 0810.0038.6180.0970.0084.71FB   ;2758  D_RC[0E],SC_K[.F]
U 11FB. 0000.003C.0580.0800.0084.B1FC   ;2759  SC_SC-K[.1]
U 11FC. 0001.2034.01C0.09E8.0000.11FD   ;2760  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU
U 11FD. 001D.0000.0180.0800.0010.11FE   ;2761  ALU_D-Q,CLK.UBCC ; CHECK RESULT
U 11FE. 0000.013C.0180.0800.0000.109E   ;2762  Z?
U 109E. 0000.003D.0180.0800.0000.115D   ;2763 =0 ERROR1
```

```
U 109F. 0810.0038.0180.0960.0010.11FF  ;2764  D_RC[0C],CLK.UBCC ; FINISHED TEST?
U 11FF. 0000.013C.0180.0800.0000.10A0  ;2765  Z?
U 10A0. 0019.0000.0580.09E0.0000.1098  ;2766 =0 ALU_D-K[.1],RC[0C]_ALU,J/TRP3A ; NO SO CONTINUE.
U 10A1. 0000.003C.0180.0800.0000.1300  ;2767  J/TRP3B
          ;2768 ; THIS TABLE IS USED TO EXECUTE ACF AND ACM FUNCTIONS WHICH
          ;2769 ; SHOULD NOT CAUSE A TRAP.
          ;2770 =00
U 1008. 0000.007E.5BF0.2C00.0000.0001  ;2771 TRP3TB: ACF/1,ACM/7,Q_ID[ACC.2],RETURN1
U 1009. 0000.00FE.59F0.2C00.0000.0001  ;2772  ACF/3,ACM/3,Q_ID[ACC.2],RETURN1
U 100A. 0000.00FE.5AF0.2C00.0000.0001  ;2773  ACF/3,ACM/5,Q_ID[ACC.2],RETURN1
U 100B. 0000.00FE.5B70.2C00.0000.0001  ;2774  ACF/3,ACM/6,Q_ID[ACC.2],RETURN1
          ;2775 ; COME HERE WHEN TEST COMPLETE
U 1300. 0000.003C.0180.0800.0000.10A2  ;2776 TRP3B: NOP
```

```
                ;2777  .PAGE "TEST 223 UBREAK REGISTER AND SYNC TEST"
                ;2778 ;*****************************************************************
                ;2779 ;**
                ;2780 ; UBREAK REGISTER AND SYNC TEST
                ;2781 ;
                ;2782 ; TEST DESCRIPTION
                ;2783 ;  THIS IS A TEST OF THE MICRO BREAK REGISTER AND OF
                ;2784 ;  THE COMPARATORS WHICH GENERATE THE SYNC ON MATCH.
                ;2785 ;  THIS COMPARATOR IS TESTED TO INSURE NOT ONLY
                ;2786 ;  VALID MATCHES BUT ALSO VALID MISMATCHES. TEST ADDRESSES
                ;2787 ;  ARE GENERATED BY FLOATING A 1 ACROSS 0'S.
                ;2788 ;  THE TEST SEQUENCE FOR EACH TEST PATTERN IS THEN:
                ;2789 ;   1 WRITE THE PATTERN INTO UBREAK REG.
                ;2790 ;   2 EXECUTE LOCATION 0 WITH NO SYNC
                ;2791 ;   3 EXECUTE THE LOCATION ADDRESSED BY
                ;2792 ;     THE PATTERN AND GET A SYNC
                ;2793 ;   4 WRITE 0 INTO THE UBREAK REG.
                ;2794 ;   5 EXECUTE THE LOCATION ADDRESSED BY
                ;2795 ;     BY THE TEST PATTERN WITH NO SYNC
                ;2796 ;   6 EXECUTE LOCATION 0 AND GET A SYNC.
                ;2797 ;
                ;2798 ; FPA UCODE USED
                ;2799 ;  LOCATION CONTENTS
                ;2800 ;  8:  NAD/100
                ;2801 ;  9:  NAD/80
                ;2802 ;  0A:  NAD/40
                ;2803 ;  0B:  NAD/20
                ;2804 ;  0C:  NAD/10
                ;2805 ;  0D:  NAD/8
                ;2806 ;  0E:  NAD/4
                ;2807 ;  0F:  NAD/2
                ;2808 ;  10:  NAD/1
                ;2809 ;  11:  NAD/0
                ;2810 ;
                ;2811 ; LOGIC DESCRIPTION
                ;2812 ;
                ;2813 ; ERROR DESCRIPTION
                ;2814 ;  EXPECTED FPA MAINTENANCE REG.
                ;2815 ;  GOT FPA MAINTENANCE REG.
                ;2816 ;  UBREAK REGISTER
                ;2817 ;
                ;2818 ; SYNC POINT DESCRIPTION
                ;2819 ;
                ;2820 ;--
                ;2821 ;*****************************************************************
                ;2822 =0
U 10A2. 0000.003D.0180.0800.0000.1149  ;2823 UBRK: NEWTST
U 10A3. 0018.0038.0D80.09F8.0000.1301  ;2824  RC[0F]_K[.3]
U 1301. 0018.0038.0180.0A80.0000.1302  ;2825  R[0]_K[.8]   ; SET UP A COUNTER.
U 1302. 0018.0038.0180.0A88.0000.10A4  ;2826  R[1]_K[.8]   ; USED TO TRAP TO FPA UCODE
U 10A4. 0000.003D.0180.0800.0000.115C  ;2827 =0 ERLOOP
U 10A5. 0000.003C.0180.0800.0000.10A6  ;2828  NOP
                ;2829 =0
U 10A6. 0000.003D.0180.0800.0000.1166  ;2830 UBRK1: CALL,J/FPINIT
U 10A7. 0000.003C.0180.0A00.0082.1303  ;2831  SC_R[0]    ; GENERATE TEST PATTERN OF 1 IN 0'S.
```

```
U 1303. 0803.001C.6180.09E0.0084.7304  ;2832  D_NOT.MASK,RC[0C]_ALU,SC_K[.F]
U 1304. 0801.001C.0180.0800.0000.1305  ;2833  ALU_D.ORNOT.MASK,D_ALU ; SET BIT 15=LOAD UBREAK REGISTER.
U 1305. 0000.003C.5980.3C00.0000.1306  ;2834  ID[ACC.2]_D
U 1306. 0818.0038.6580.0800.0000.10A8  ;2835  D_K[.10]  ; TRAP TO LOCATION 11 WHICH
U 10A8. 0819.0015.0580.0800.0000.116B  ;2836 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA ; WILL JUMP TO 0.
U 10A9. 0000.003C.5980.3C00.0000.1307  ;2837  ID[ACC.2]_D
U 1307. 0000.00FC.0380.0800.0000.1308  ;2838  TRAP.FPA
U 1308. 0000.003C.59F0.2C00.0000.1309  ;2839  Q_ID[ACC.2]  ; CHECK SYNC BIT.
U 1309. 0001.203C.8D80.09E8.0084.730A  ;2840  RC[0D]_Q,SC_K[.1F]
U 130A. 0801.0034.0180.09F0.0000.130B  ;2841  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU
U 130B. 001D.0000.0180.0800.0010.130C  ;2842  ALU_D-Q,CLK.UBCC
U 130C. 0000.013C.0180.0800.0000.10AA  ;2843  Z?
U 10AA. 0000.003D.0180.0800.0000.115D  ;2844 =0 ERROR1   ; GOT A SYNC OR MAINTENANCE REGISTER
          ;2845      ; BROKEN.
U 10AB. 0800.003C.0180.0A08.0000.10AC  ;2846  D_R[1]   ; GO TO THE UCODE LOCATION
U 10AC. 0000.003D.0180.0800.0000.116B  ;2847 =0 CALL,J/GENTRA  ; WHICH IS ADDRESSED BY THE FLOATING
U 10AD. 0000.003C.5980.3E00.0082.130D  ;2848  ID[ACC.2]_D,SC_R[0] ; PATTERN.
U 130D. 0801.001C.6180.0800.0084.730E  ;2849  ALU_D.ORNOT.MASK,D_ALU,SC_K[.F]
U 130E. 0000.00FC.0780.0800.0084.B30F  ;2850  TRAP.FPA,SC_SC-K[.1]
U 130F. 0801.001C.59F0.2C00.0000.1310  ;2851  Q_ID[ACC.2],ALU_D.ORNOT.MASK,D_ALU ; SHOULD SYNC HERE
U 1310. 0001.203C.8D80.09E8.0084.7311  ;2852  RC[0D]_Q,SC_K[.1F]
U 1311. 0801.0034.0180.09F0.0000.1312  ;2853  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU
U 1312. 001D.0000.0180.0800.0010.1313  ;2854  ALU_D-Q,CLK.UBCC
U 1313. 0000.013C.0180.0800.0000.10AE  ;2855  Z?
U 10AE. 0000.003D.0180.0800.0000.115D  ;2856 =0 ERROR1   ; DID NOT SYNC
U 10AF. 0000.003C.0180.0800.0000.10B0  ;2857  NOP
U 10B0. 0000.003D.0180.0800.0000.1166  ;2858 =0 CALL,J/FPINIT
U 10B1. 0000.003C.6180.0800.0084.7314  ;2859  SC_K[.F]  ; SET UBREAK REGISTER TO 0
U 1314. 0803.001C.0180.0800.0000.1315  ;2860  D_NOT.MASK
U 1315. 0003.003C.5980.3DE0.0000.1316  ;2861  RC[0C]_0,ID[ACC.2]_D
U 1316. 0800.003C.0180.0A08.0000.10B2  ;2862  D_R[1]
U 10B2. 0000.003D.0180.0800.0000.116B  ;2863 =0 CALL,J/GENTRA  ; TRAP TO THE LOCATION ADDRESSED
U 10B3. 0000.003C.5980.3E00.0082.1317  ;2864  ID[ACC.2]_D,SC_R[0] ; BY THE FLOATING PATTERN.
U 1317. 0801.00DC.0380.0800.0000.1318  ;2865  TRAP.FPA,ALU_D.ORNOT.MASK,D_ALU
U 1318. 0000.003C.59F0.2C00.0000.1319  ;2866  Q_ID[ACC.2]  ; SYNC SHOULD NOT BE ASSERTED.
U 1319. 0001.203C.8D80.09E8.0084.731A  ;2867  RC[0D]_Q,SC_K[.1F]
U 131A. 0801.0034.0180.09F0.0000.131B  ;2868  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU
U 131B. 001D.0000.0180.0800.0010.131C  ;2869  ALU_D-Q,CLK.UBCC
U 131C. 0000.013C.0180.0800.0000.10B4  ;2870  Z?
U 10B4. 0000.003D.0180.0800.0000.115D  ;2871 =0 ERROR1   ; UBREAK SYNC'ED ON MISMATCH OR
U 10B5. 0000.003C.0180.0800.0000.10B6  ;2872  NOP   ; MAINTENANCE REGISTER BROKEN.
U 10B6. 0000.003D.0180.0800.0000.1166  ;2873 =0 CALL,J/FPINIT
U 10B7. 0818.0038.6580.0800.0000.10B8  ;2874  D_K[.10]  ; EXECUTE LOCATION 0 TO GET SYNC.
U 10B8. 0819.0015.0580.0800.0000.116B  ;2875 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 10B9. 0000.003C.5980.3C00.0000.131D  ;2876  ID[ACC.2]_D
U 131D. 0000.003C.6180.0800.0084.731E  ;2877  SC_K[.F]
U 131E. 0000.003C.0580.0800.0084.B31F  ;2878  SC_SC-K[.1]
U 131F. 0801.00DC.0380.0800.0000.1320  ;2879  TRAP.FPA,ALU_D.ORNOT.MASK,D_ALU
U 1320. 0000.003C.59F0.2C00.0000.1321  ;2880  Q_ID[ACC.2]  ; SHOULD HAVE SYNC HERE.
U 1321. 0001.203C.8D80.09E8.0084.7322  ;2881  RC[0D]_Q,SC_K[.1F]
U 1322. 0801.0034.0180.09F0.0000.1323  ;2882  ALU_D.AND.MASK,D_ALU,RC[0E]_ALU
U 1323. 001D.0000.0180.0800.0010.1324  ;2883  ALU_D-Q,CLK.UBCC
U 1324. 0000.013C.0180.0800.0000.10BA  ;2884  Z?
U 10BA. 0000.003D.0180.0800.0000.115D  ;2885 =0 ERROR1   ; NO SYNC ASSERTED OR MAINTENANCE
          ;2886      ; REGISTER BROKEN.
```

```
U 10BB. 0000.003C.0180.0A08.0000.1325  ;2887  LAB_R[1]  ; INCREMENT ADDRESS USED TO
U 1325. 0018.0014.0580.0A88.0000.1326  ;2888  R[1]_LA+K[.1]  ; GET TO FPA UCODE.
U 1326. 0000.003C.0180.0A00.0010.1327  ;2889  ALU_R[0],CLK.UBCC ; TRIED ALL PATTERNS?
U 1327. 0000.013C.0180.0800.0000.10BC  ;2890  Z?
U 10BC. 0018.0000.0580.0A80.0000.10A6  ;2891 =0 R[0]_LA-K[.1],J/UBRK1 ; IF NOT CONTINUE TEST.
U 10BD. 0000.003C.0180.0800.0000.10BE  ;2892  NOP  ; DONE
```

```
                ;2893  .PAGE "TEST 224 TEST FOR FPA IN IRD STATE"
                ;2894  ;*****************************************************************
                ;2895  ;**
                ;2896  ; TEST IF FPA IS IN THE IRD STATE.
                ;2897  ;
                ;2898  ; TEST DESCRIPTION
                ;2899  ; THE FPA IS INITIALIZED BY FPINIT. THIS WILL HOLD THE
                ;2900  ; FPA AT UADDRESS 028 AFTER LOADING A HALT INTO THE
                ;2901  ; INSTRUCTION BUFFER. A TRAP TO UADDRESS 0 PUTS THE FPA
                ;2902  ; INTO ITS IRD STATE. THE FPA WILL THEN IDLE IN IRD.
                ;2903  ; THE NEXT UADDRESS IS READ TO DETERMINE IF THE FPA IS
                ;2904  ; IN IRD.
                ;2905  ;
                ;2906  ; FPA UCODE USED
                ;2907  ; LOCATION CONTENTS
                ;2908  ; 000: BSC/NH,FADC/LD,SGNC/A.RES,MSC/RR.0,OPLD/INIT,EALUC/3,NAD/0A3
                ;2909  ; 0A3: BSC/R,FADC/R1,SGNC,LDS,FPSYNC,OPLD/R.R,EAC/LDAB,MSC/IR0,SCR/R.R,NAD/180
                ;2910  ;
                ;2911  ; LOGIC DESCRIPTION
                ;2912  ; THE FPA USEQUENCER LOGIC IS ON THE FCT MODULE.
                ;2913  ;
                ;2914  ; ERROR DESCRIPTION
                ;2915  ; EXPECTED NEXT UADDRESS
                ;2916  ; RECEIVED NEXT UADDRESS
                ;2917  ;
                ;2918  ; SYNC POINT DESCRIPTION
                ;2919  ;--
                ;2920  ;*****************************************************************
                ;2921  ;/////////////////////////////////////////////
                ;2922  ;*
                ;2923  ; TEST FOR FPA IN IRD
                ;2924  ;-
                ;2925  =0
U 10BE. 0018.0039,0980.09F8.0000,1149  ;2926 BTST3: NEWTST[.2]
U 10BF. 0000.003C,0180.0800.0000,10C0  ;2927  NOP
U 10C0. 0000.003D,0180.0800.0000,1166  ;2928 =0 CALL[FPINIT]
U 10C1. 0F00.003C,0180.0800.0000,10C2  ;2929  D_0
U 10C2. 0000.003D,0180.0800.0000,11A7  ;2930 =0 CALL[TRAP]  ; TRAP TO ADDRESS ZERO
U 10C3. 0000.003C,59F0.2C00.0000,1328  ;2931  Q_ID[ACC.2]  ; READ FPA MAINTENANCE REGISTER
U 1328. 0818.0038,4180.0800.0000,1329  ;2932  D_K[.80]
U 1329. 0500.003C,0180.0800.0000,132A  ;2933  D_D.LEFT
U 132A. 0819.0030,4180.0800.0000,132B  ;2934  D_D.OR.K[.80]  ; EXPECTED ADDRESS (180)
U 132B. 0001.003C,0180.09F0.0000,132C  ;2935  RC[0E]_D  ; SAVE
U 132C. 0818.0038,4980.0800.0000,132D  ;2936  D_K[.FF]
U 132D. 0500.003C,0180.0800.0000,132E  ;2937  D_D.LEFT
U 132E. 0819.0030,0580.0800.0000,132F  ;2938  D_D.OR.K[.1]  ; MASK LOW 9 BITS
U 132F. 001D.0034,01C0.0800.0000,1330  ;2939  ALU_D.AND.Q.Q_ALU ; PASS ADDRESS BITS ONLY
U 1330. 0001.203C,0180.09E8.0000,1331  ;2940  RC[0D]_Q  ; SAVE
U 1331. 0810.0038,0180.0970.0000,1332  ;2941  D_RC[0E]
U 1332. 001D.0020,0180.0800.0010,1333  ;2942  ALU_D.XOR.Q.CLK.UBCC
U 1333. 0000.013C,0180.0800.0000,10C4  ;2943  Z?
U 10C4. 0000.003D,0180.0800.0000,115D  ;2944 =0 ERROR1  ; NOT IN IRD
U 10C5. 0000.003C,0180.0800.0000,10C6  ;2945 ENTST2: NOP
```

```
                ;2946  .PAGE "TEST 225 TEST FPA/CPU SYNC AND WAIT HAND SHAKE"
                ;2947 ;*****************************************************************
                ;2948 ;++
                ;2949 ; TEST FPA/CPU SYNC AND WAIT HAND SHAKE
                ;2950 ;
                ;2951 ; TEST DESCRIPTION
                ;2952 ; THE FPA IS INITIALIZED AND THEN TRAPPED TO UADDRESS 08F
                ;2953 ; AT 08F WAIT IS ASSERTED, AND FPSYNC IS DEASSERTED. FPSYNC
                ;2954 ; (ACCELERATOR SYNC) IS VERIFIED DEASSERTED. CPSYNC IS
                ;2955 ; ISSUED FROM THE CPU TO STEP THE FPA TO THE NEXT UADDRESS
                ;2956 ; (0FA) FPSYNC IS VERIFIED ASSERTED AND CPSYNC IS AGAIN
                ;2957 ; ISSUED FROM THE CPU. AT THE NEXT UADDRESS (0FB) FPSYNC
                ;2958 ; IS VERIFIED DEASSERTED.
                ;2959 ;
                ;2960 ; FPA UCODE USED
                ;2961 ; LOCATION CONTENTS
                ;2962 ; 08F: BSC/ID,FADC/LD,WAIT,NAD/0FA
                ;2963 ; 0FA: BSC/ID,FADC/AR,WAIT,FPSYNC,NAD/0FB
                ;2964 ; 0FB: BSC/FAL.HL,FADC/A,WAIT,NAD/0FC
                ;2965 ;
                ;2966 ; LOGIC DESCRIPTION
                ;2967 ; THE FPA USEQUENCER LOGIC IS ON THE FCT MODULE.
                ;2968 ;
                ;2969 ; ERROR DESCRIPTION
                ;2970 ; EXPECTED NEXT UADDRESS
                ;2971 ; RECEIVED NEXT UADDRESS
                ;2972 ;
                ;2973 ; SYNC POINT DESCRIPTION
                ;2974 ;--
                ;2975 ;*****************************************************************
                ;2976 ;/////////////////////////////////////////
                ;2977 ;+
                ;2978 ; TEST FPA/CPU SYNC AND WAIT HAND SHAKE
                ;2979 ;-
                ;2980 =0
U 10C6, 0018.0039.0980.09F8.0000.1149  ;2981 BTST4: NEWTST[.2]
U 10C7, 0000.003C.0180.0800.0000.10C8  ;2982  NOP
U 10C8, 0000.003D.0180.0800.0000.1166  ;2983 =0 CALL[FPINIT]
U 10C9, 0000.003C.0180.0800.0000.1334  ;2984  NOP
U 1334, 0818.0038.CD80.0800.0000.1335  ;2985  D_K[.F0]
U 1335, 0819.0030.8580.0800.0000.1336  ;2986  D_D.OR.K[.C]
U 1336, 0001.003C.0180.09F0.0000.10CA  ;2987  RC[0E]_D ; EXPECTED NEXT UADDRESS
U 10CA, 0000.003D.0180.0800.0000.115C  ;2988 =0 ERLOOP
U 10CB, 0000.003C.0180.0800.0000.10CC  ;2989  NOP
U 10CC, 0000.003D.0180.0800.0000.11B2  ;2990 =0 CALL[TRP.8F] ; TRAP TO 08F
U 10CD, 0000.063C.0180.0800.0000.1026  ;2991  ACC.SYNC? ; AT FPA UADDRESS 08F
U 1026, 0000.007C.0180.0800.0000.10CF  ;2992 =110 CPSYNC,J/SYNCHK ; OK ACKNOWLEDGED NO SYNC AT 08F
U 1027, 0000.003C.59F0.2C00.0000.1337  ;2993  Q_ID[ACC.2] ; ERROR
U 1337, 0001.203C.0180.09E8.0000.10CE  ;2994  RC[0D]_Q ; SAVE RECEIVED NEXT ADDRESS
U 10CE, 0000.003D.0180.0800.0000.115D  ;2995 =0 ERROR1
U 10CF, 0000.063C.0180.0800.0000.1036  ;2996 SYNCHK: ACC.SYNC? ; AT FPA UADDRESS 0FA
U 1036, 0000.003C.0180.0800.0000.1047  ;2997 =110 J/ERCALL ; ERROR DID NOT ACKNOWLEDGE SYNC AT 0FA
U 1037, 0000.007C.0180.0800.0000.1338  ;2998  CPSYNC  ; OK ACKNOWLEDGED SYNC AT 0FA
U 1338, 0000.063C.0180.0800.0000.1046  ;2999  ACC.SYNC? ; AT FPA UADDRESS 0FB
U 1046, 0000.003C.0180.0800.0000.10D1  ;3000 =110 J/ENTST3 ; OK ACKNOWLEDGED NO SYNC AT 0FB
```

```
U 1047. 0000.003C.59F0.2C00.0000.1339  ;3001 ERCALL: Q_ID[ACC.2]
U 1339. 0001.203C.0180.09E8.0000.10D0  ;3002  RC[0D]_Q  ; SAVE
U 10D0. 0000.003D.0180.0800.0000.115D  ;3003 =0 ERROR1
U 10D1. 0000.003C.0180.0800.0000.10D2  ;3004 ENTST3: NOP
```

```
          ;3005  .PAGE "TEST 226 ACCELERATER INSTRUCTION TEST"
          ;3006 ;*****************************************************************
          ;3007 ;**
          ;3008 ; ACCELERATER INSTRUCTION TEST
          ;3009 ;
          ;3010 ; TEST DESCRIPTION
          ;3011 ;  THIS IS A TEST OF THE ACCELERATER'S RESPONSE
          ;3012 ;  TO ALL 8 BIT OP CODES WHILE IN IRD. THE ONLY
          ;3013 ;  CHECK MADE HERE IS TO INSURE THAT THE FPA
          ;3014 ;  LEAVES THE IRD STATE WHEN THE OP CODE IS A FLOATING
          ;3015 ;  POINT INSTRUCTION AND STAYS IN THE IRD STATE WHEN
          ;3016 ;  THE OPCODE IS NOT A FLOATING POINT INSTRUCTION.
          ;3017 ;  A COUNT PATTERN IS RUN THRU THE 8 BIT OPCODE
          ;3018 ;  AND EACH IS TRIED.
          ;3019 ;
          ;3020 ; FPA UCODE USED
          ;3021 ;  LOCATION CONTENTS
          ;3022 ;  0:  IRD.TRP,NAD/IRD.1
          ;3023 ;  IRD.1:
          ;3024 ;  103:  FPA.IRD,FPSYNC,NAD/180
          ;3025 ;
          ;3026 ; LOGIC DESCRIPTION
          ;3027 ;
          ;3028 ; ERROR DESCRIPTION
          ;3029 ;  OPCODE BEING TESTED
          ;3030 ;  GOT FPA MAINTENANCE REGISTER
          ;3031 ;   NOTE THAT THIS HAS ONE OF TWO SIGNIFICANCES:
          ;3032 ;   1 THE OPCODE WAS A FLOATING POINT OPCODE
          ;3033 ;    AND THE FPA SHOULD HAVE LEFT IRD
          ;3034 ;   OR
          ;3035 ;   2 THE OPCODE WAS NOT A FLOATING POINT
          ;3036 ;    OPCODE AND THE FPA SHOULD HAVE STAYED
          ;3037 ;    IN IRD.
          ;3038 ;
          ;3039 ; SYNC POINT DESCRIPTION
          ;3040 ;
          ;3041 ;--
          ;3042 ;*****************************************************************
          ;3043 =0
U 10D2. 0000.003D.0180.0800.0000.1149  ;3044 ACCIN: NEWTST
U 10D3. 0018.0038.0980.09F8.0000.133A  ;3045  RC[0F]_K[.2]
U 133A. 0018.0038.4980.09F0.0000.133B  ;3046  RC[0E]_K[.FF]  ; INITIALIZE OPCODE
U 133B. 0818.0038.7DF8.0800.0000.133C  ;3047  D_K[.1B],Q_0
U 133C. 0000.003C.1180.0800.0084.733D  ;3048  SC_K[.4]
U 133D. 0D00.003C.0180.0800.0000.10D4  ;3049  D_DAL.SC
U 10D4. 0001.003D.0180.0A88.0000.115C  ;3050 =0 R[1]_D,ERLOOP
U 10D5. 0000.003C.0180.0800.0000.10D6  ;3051  NOP
          ;3052 =0
U 10D6. 0000.003D.0180.0800.0000.1166  ;3053 ACCIN1: CALL,J/FPINIT
U 10D7. 0000.003C.0180.0800.0000.10D8  ;3054  NOP
U 10D8. 0000.003D.0180.0800.0000.102C  ;3055 =0 CALL,J/FPIRD  ; PUT FPA IN ITS IRD STATE
U 10D9. 0018.0038.0180.0800.0200.133E  ;3056  VA_K[.8]
U 133E. 0810.0038.0180.0970.0000.10DA  ;3057  D_RC[0E]
U 10DA. 0000.803D.0180.A800.0000.117B  ;3058 =0 MCT/WRITE.P,DT/BYTE,CALL,J/LOADIB ; WRITE OPCODE
          ;3059     ; INTO CACHE AND GO LOAD
```

```
          ;3060     ; IT INTO THE INSTRUCTION BUFFER.
U 10DB. 0000.003C.0180.0800.0000.1020  ;3061  NOP
U 1020. 0000.003D.0180.0800.0000.1021  ;3062 =00 CALL
U 1021. 0000.003F.59F0.2C00.0000.1200  ;3063  Q_ID[ACC.2],J/IRTAB,SUB/SPEC ; GET FPA MAINTENANCE
          ;3064     ; REGISTER AND ISSUE CALL/3.
          ;3065     ; SAVE MAINTENANCE REGISTER WHICH
          ;3066     ; WILL INDICATE WHETHER OR NOT
          ;3067     ; THE FPA STAYED IN IRD BY ITS NAD
          ;3068     ; FIELD.
          ;3069 ; NOW SEE IF THE OPCODE WAS A FLOATING OPCODE:
U 1022. 0019.0034.0180.0800.0010.1023  ;3070  ALU_D.AND.K[.8],CLK.UBCC
U 1023. 0000.013C.0180.0800.0000.10DC  ;3071  Z?
U 10DC. 0000.003C.0180.0800.0000.134C  ;3072 =0 J/NOFPIN   ; NOT FPA OPCODE
U 10DD. 0019.0034.3180.0800.0010.133F  ;3073  ALU_D.AND.K[.40],CLK.UBCC
U 133F. 0000.013C.0180.0800.0000.10DE  ;3074  Z?
U 10DE. 0018.0038.41C0.0800.0000.1340  ;3075 =0 J/ACCIN2,Q_K[.80]
U 10DF. 0000.003C.0180.0800.0000.134C  ;3076  J/NOFPIN   ; NOT FPA OPCODE
U 1340. 0019.2014.09C0.0800.0000.1341  ;3077 ACCIN2: Q_Q+K[.2]
U 1341. 001D.0034.0180.0800.0010.1342  ;3078  ALU_D[AND]Q,CLK.UBCC
U 1342. 0000.013C.0180.0800.0000.10E0  ;3079  Z?
U 10E0. 0000.003C.0180.0800.0000.10E3  ;3080 =0 J/ACCIN3
U 10E1. 0019.0034.1180.0800.0010.1343  ;3081  ALU_D.AND.K[.4],CLK.UBCC
U 1343. 0000.013C.0180.0800.0000.10E2  ;3082  Z?
U 10E2. 0000.003C.0180.0800.0000.1352  ;3083 =0 J/FPIN    ; FPA OPCODE
U 10E3. 0018.0038.41C0.0800.0000.1344  ;3084 ACCIN3: Q_K[.80]
U 1344. 0018.0038.6580.0800.0000.1345  ;3085  ALU_K[.10]
U 1345. 0019.2014.65C0.0800.0000.1346  ;3086  Q_Q+K[.10]
U 1346. 001D.0034.0180.0800.0010.1347  ;3087  ALU_D[AND]Q,CLK.UBCC
U 1347. 0000.013C.0180.0800.0000.10E4  ;3088  Z?
U 10E4. 0018.0038.79C0.0800.0000.1348  ;3089 =0 J/ACCIN4,Q_K[.30]
U 10E5. 0000.003C.0180.0800.0000.1352  ;3090  J/FPIN    ; FPA OPCODE
U 1348. 0019.2014.09C0.0800.0000.1349  ;3091 ACCIN4: Q_Q+K[.2]
U 1349. 001D.0034.0180.0800.0010.134A  ;3092  ALU_D[AND]Q,CLK.UBCC
U 134A. 0000.013C.0180.0800.0000.10E6  ;3093  Z?
U 10E6. 0000.003C.0180.0800.0000.134C  ;3094 =0 J/NOFPIN   ; NOT FPA OPCODE
U 10E7. 0019.0034.1180.0800.0010.134B  ;3095  ALU_D.AND.K[.4],CLK.UBCC
U 134B. 0000.013C.0180.0800.0000.10E8  ;3096  Z?
U 10E8. 0000.003C.0180.0800.0000.1352  ;3097 =0 J/FPIN    ; FPA OPCODE
U 10E9. 0000.003C.0180.0800.0000.134C  ;3098  J/NOFPIN   ; NOT FPA OPCODE
          ;3099 ; IF THE OPCODE BEING TESTED WAS NOT AN FPA OPCODE
          ;3100 ; COME HERE TO INSURE THAT THE FPA NEVER LEFT IRD:
U 134C. 0800.003C.0180.0A08.0000.134D  ;3101 NOFPIN: D_R[1]
U 134D. 0010.0038.61C0.0968.0084.734E  ;3102  Q_RC[0D],SC_K[.F]
U 134E. 0000.003C.0580.0800.0084.B34F  ;3103  SC_SC-K[.1]
U 134F. 0001.2034.01C0.09E8.0000.1350  ;3104  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU
U 1350. 001D.0000.0180.0800.0010.1351  ;3105  ALU_D-Q,CLK.UBCC
U 1351. 0000.013C.0180.0800.0000.10EA  ;3106  Z?
U 10EA. 0000.003D.0180.0800.0000.115D  ;3107 =0 ERROR1    ; THE FPA USEQUENCER
          ;3108     ; LEFT ITS IRD STATE
          ;3109     ; WITH A NON FPA OPCODE
          ;3110     ; IN THE INSTRUCTION BUFFER.
U 10EB. 0000.003C.0180.0800.0000.1358  ;3111  J/ACCIN5
          ;3112 ; IF THE OPCODE IS  AN FPA OPCODE COME HERE TO MAKE
          ;3113 ; SURE THAT THE FPA USEQUENCER DID IN FACT LEAVE ITS IRD STATE:
U 1352. 0800.003C.0180.0A08.0000.1353  ;3114 FPIN: D_R[1]
```

```
U 1353. 0010.0038.61C0.0968.0084.7354  ;3115  Q_RC[0D],SC_K[.F]
U 1354. 0000.003C.0580.0800.0084.B355  ;3116  SC_SC-K[.1]
U 1355. 0001.2034.01C0.09E8.0000.1356  ;3117  ALU_Q.AND.MASK,Q_ALU,RC[0D]_ALU
U 1356. 001D.0000.0180.0800.0010.1357  ;3118  ALU_D-Q,CLK.UBCC
U 1357. 0000.013C.0180.0800.0000.10EC  ;3119  Z?
U 10EC. 0000.003C.0180.0800.0000.1358  ;3120 =0 J/ACCIN5
U 10ED. 0000.003D.0180.0800.0000.115D  ;3121  ERROR1   ; THE FPA STAYED IN IRD
            ;3122      ; WITH AN FPA OPCODE IN THE
            ;3123      ; INSTRUCTION BUFFER.
            ;3124 ; COME HERE TO TEST NEXT OPCODE.
U 1358. 0810.0038.0180.0970.0010.1359  ;3125 ACCIN5: D_RC[0E],CLK.UBCC ; TEST DONE?
U 1359. 0000.013C.0180.0800.0000.10EE  ;3126  Z?
U 10EE. 0019.0000.0580.09F0.0000.10D6  ;3127 =0 ALU_D-K[.1],RC[0E]_ALU,J/ACCIN1
U 10EF. 0000.003C.0180.0800.0000.10F0  ;3128  NOP
            ;3129 ; THESE LOCATIONS ARE USED BY THE PRECEDING TEST IN THE CACHE:
            ;3130 ;x8
            ;3131 ;$ 50FF.0050
            ;3132 ;
```

```
                ;3133  .PAGE "TEST 227 BEN/1 TEST"
                ;3134 ;*****************************************************************
                ;3135 ;++
                ;3136 ; BEN/1 TEST
                ;3137 ;
                ;3138 ; TEST DESCRIPTION
                ;3139 ;  THIS IS A TEST OF BEN/1. BEN/1 IS USED TO DECODE
                ;3140 ;  THE INSTRUCTION. BIT 2 IS ASSERTED WHEN THE OPCODE
                ;3141 ;  IS A FLOATING POINT (INSTEAD OF DOUBLE) INSTRUCTION,
                ;3142 ;  THIS IS MERELY AN INVERTED COPY OF BIT 5 OF THE OPCODE.
                ;3143 ;  BITS<1:0> DECODE THE INSTRUCTION TYPE.
                ;3144 ;
                ;3145 ; FPA UCODE USED
                ;3146 ;  LOCATION CONTENTS
                ;3147 ;  2A:  BEN/1,NAD/0
                ;3148 ;
                ;3149 ; LOGIC DESCRIPTION
                ;3150 ;
                ;3151 ; ERROR DESCRIPTION
                ;3152 ;  EXPECTED NEXT ADDRESS FOLLOWING EXECUTION OF 2A
                ;3153 ;  GOT NEXT ADDRESS
                ;3154 ;  OPCODE LOADED IN THE INSTRUCTION BUFFER
                ;3155 ;  CACHE ADDRESS OF OPCODE
                ;3156 ;
                ;3157 ; SYNC POINT DESCRIPTION
                ;3158 ;
                ;3159 ;--
                ;3160 ;*****************************************************************
                ;3161 =0
U 10F0, 0000.003D.0180.0800.0000.1149  ;3162 BENT1: NEWTST
U 10F1, 0018.0038.1180.09F8.0000.135A  ;3163  RC[0F]_K[.4]
U 135A, 0018.0038.8580.09D8.0000.135B  ;3164  RC[0B]_K[.C]  ; CACHE ADDRESS OF FIRST OPCODE.
U 135B, 0018.0038.5D80.09F0.0000.10F2  ;3165  RC[0E]_K[.7]  ; SET UP A COUNTER, AND
                ;3166      ; EXPECTED NAD RESULT.
U 10F2, 0000.003D.0180.0800.0000.115C  ;3167 =0 ERLOOP
U 10F3, 0000.003C.0180.0800.0000.10F4  ;3168  NOP
                ;3169 =0
U 10F4, 0000.003D.0180.0800.0000.1166  ;3170 BENT1A: CALL,J/FPINIT
U 10F5, 0010.0038.0180.0958.0200.135C  ;3171  VA_RC[0B]  ; LOAD OPCODE ADDRESS INTO VA.
U 135C, 0000.003C.0180.4000.0000.135D  ;3172  D[LONG]_CACHE  ; GET OPCODE.
U 135D, 0001.003C.0180.09E0.0000.10F6  ;3173  RC[0C]_D  ; SAVE IT FOR ERROR REPORT.
U 10F6, 0000.003D.0180.0800.0000.117B  ;3174 =0 CALL,J/LOADIB  ; LOAD OPCODE INTO IB.
U 10F7, 0818.0038.2D80.0800.0000.10F8  ;3175  D_K[.28]
U 10F8, 0819.0015.0980.0800.0000.116B  ;3176 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; GENERATE ADDRESS OF
                ;3177      ; FPA UCODE WITH BEN/1.
U 10F9, 0000.003C.5980.3C00.0000.135E  ;3178  ID[ACC.2]_D
U 135E, 0810.00F8.0380.0970.0000.135F  ;3179  TRAP,FPA,D_RC[0E] ; TRAP TO FPA LOCATION 2A
U 135F, 0000.003C.59F0.2C00.0000.1360  ;3180  Q_ID[ACC.2]  ; GET THE NEXT ADDRESS TO BE
U 1360, 0019.2034.81C0.09E8.0000.1361  ;3181  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU ; EXECUTED.
U 1361, 001D.0000.0180.0800.0010.1362  ;3182  ALU_D-Q,CLK.UBCC
U 1362, 0000.013C.0180.0800.0000.10FA  ;3183  Z?
U 10FA, 0000.003D.0180.0800.0000.115D  ;3184 =0 ERROR1    ; BEN/1 FAILED
U 10FB, 0810.0038.0180.0958.0000.1363  ;3185  D_RC[0B]  ; INCREMENT POINT TO THEE
U 1363, 0019.0014.1180.09D8.0000.1364  ;3186  ALU_D+K[.4],RC[0B]_ALU ; NEXT OPCODE.
U 1364, 0010.0038.01C0.0960.0000.1365  ;3187  Q_RC[0C]
```

```
U 1365. 0019.2034.49C0.0800.0000.1366  ;3188  ALU_Q[AND]K[.FF],Q_ALU ; COMPUTE NEXT BEN ADDRESS
U 1366. 0818.0038.D180.0800.0000.1367  ;3189  D_K[.C0]
U 1367. 0819.0014.1180.0800.0000.1368  ;3190  ALU_D+K[.4],D_ALU
U 1368. 001D.0000.0180.0800.0010.1369  ;3191  ALU_D-Q,CLK.UBCC
U 1369. 0000.013C.0180.0800.0000.10FC  ;3192  Z?
U 10FC. 0000.003C.0180.0800.0000.136A  ;3193 =0 J/BENT1B
U 10FD. 0000.003C.0180.0800.0000.10F4  ;3194  J/BENT1A
U 136A. 0810.0038.0180.0970.0010.136B  ;3195 BENT1B: D_RC[0E],CLK.UBCC ; IS THE TEST DONE.
U 136B. 0000.013C.0180.0800.0000.10FE  ;3196  Z?
U 10FE. 0019.0000.0580.09F0.0000.10F4  ;3197 =0 RC[0E]_ALU,ALU_D-K[.1],J/BENT1A
U 10FF. 0000.003C.0180.0800.0000.110A  ;3198  NOP
            ;3199 ; THE FOLLOWING DATA IS TO BE USED BY THE PRECEDING TEST:
            ;3200 ;
            ;3201 ;%0C
            ;3202 ; MULL
            ;3203 ;$ 50C4,0050
            ;3204 ; EMODF
            ;3205 ;$ 5054,0050
            ;3206 ; POLYF
            ;3207 ;$ 5055,0050
            ;3208 ; MULF3
            ;3209 ;$ 5045,0050
            ;3210 ; DIVF2
            ;3211 ;$ 5046,0050
            ;3212 ; EMODD
            ;3213 ;$ 5074,0050
            ;3214 ; ADDD2
            ;3215 ;$ 5060,0050
            ;3216 ; MULD3
            ;3217 ;$ 5065,0050
            ;3218 ; DIVD2
            ;3219 ;$ 5066,0050
            ;3220 ;
```

```
                ;3221   .PAGE "TEST 228 A-FORK TEST"
                ;3222  ;*****************************************************************
                ;3223  ;**
                ;3224  ; A-FORK TEST
                ;3225  ;
                ;3226  ; TEST DESCRIPTION
                ;3227  ;  THIS IS A TEST OF THE LOGIC WHICH DETERMINES THE
                ;3228  ;  ADDRESS  GENERATED DURING A-FORK IN THE FPA
                ;3229  ;  USEQUENCER. THIS TEST MAKES USE OF TABLE IN THE
                ;3230  ;  CACHE WHICH CONTAINS OPCODES LOADED INTO THE IB
                ;3231  ;  WHEN A-FORK IS INITIATED. SPECIFICALLY THIS TABLE
                ;3232  ;  IS MADE UP OF:
                ;3233  ;    1 OPCODE COUNT
                ;3234  ;    2 OPCODE FOLLOWED BY ITS A-FORK ADDRESS
                ;3235  ;    3 OPCODE FOLLOWED BY ITS A-FORK ADDRESS
                ;3236  ;    ...
                ;3237  ;    N OPCODE FOLLOWED BY ITS A-FORK ADDRESS
                ;3238  ;
                ;3239  ; FPA UCODE USED
                ;3240  ;  LOCATION CONTENTS
                ;3241  ;  0:  IRD.TRP,NAD/IRD.1
                ;3242  ;  IRD.1:
                ;3243  ;  103:  FPA.IRD,FPSYNC,NAD/180
                ;3244  ;
                ;3245  ; LOGIC DESCRIPTION
                ;3246  ;
                ;3247  ; ERROR DESCRIPTION
                ;3248  ;  EXPECTED A-FORK ADDRESS
                ;3249  ;  GOT A-FORK ADDRESS
                ;3250  ;  OPCODE LOADED IN IB DURING A-FORK
                ;3251  ;  ADDRESS IN CACHE OF THAT OPCODE
                ;3252  ;  COUNTER
                ;3253  ;
                ;3254  ; SYNC POINT DESCRIPTION
                ;3255  ;
                ;3256  ;--
                ;3257  ;*****************************************************************
                ;3258  =0
U 110A, 0000,003D,0180,0800,0000,1149   ;3259 AFORK: NEWTST
U 110B, 001B,0010,1180,09F8,0000,136C   ;3260  ALU_0+K[.4]+1,RC[0F]_ALU
U 136C, 0018,0038,79C0,0800,0200,136D   ;3261  VA_K[.30],Q_K[.30] ; GET OPCODE COUNT
U 136D, 0000,003C,0180,4000,0000,136E   ;3262  D[LONG]_CACHE
U 136E, 0001,003C,0180,09D0,0000,136F   ;3263  RC[0A]_D
U 136F, 0019,2014,0180,09D8,0000,1110   ;3264  ALU_Q+K[.8],RC[0B]_ALU ; INITIALIZE OPCODE ADDRESS.
U 1110, 0000,003D,0180,0800,0000,115C   ;3265 =0 ERLOOP
U 1111, 0000,003C,0180,0800,0000,1112   ;3266  NOP
                ;3267 =0
U 1112, 0000,003D,0180,0800,0000,1166   ;3268 AFORK1: CALL,J/FPINIT
U 1113, 0000,003C,0180,0800,0000,1114   ;3269  NOP
U 1114, 0000,003D,0180,0800,0000,102C   ;3270 =0 CALL,J/FPIRD
U 1115, 0010,0038,0180,0958,0200,1370   ;3271  VA_RC[0B]  ; GET OPCODE AND
U 1370, 0000,003C,0180,4000,0000,1371   ;3272  D[LONG]_CACHE
U 1371, 0001,003C,0180,09E0,0000,1116   ;3273  RC[0C]_D
U 1116, 0000,003D,0180,0800,0000,117B   ;3274 =0 CALL,J/LOADIB  ; LOAD IT INTO THE IB.
U 1117, 0000,003C,0180,0800,0000,1030   ;3275  NOP
```

```
U 1030, 0000.003D.0180.0800.0000.1031  ;3276 =00 CALL
U 1031, 0000.003F.59F0.2C00.0000.1200  ;3277  Q_ID[ACC.2],J/IRTAB,SUB/SPEC ; TEST INSTRUCTION
U 1032, 0810.0038.0180.0958.0000.1033  ;3278  D_RC[0B]  ; GET A-FORK EXPECTED ADDRESS
U 1033, 0F19.0014.1180.0800.0200.1372  ;3279  ALU_D+K[.4],VA_ALU,D_0 ; FROM TABLE IN THE CACHE.
U 1372, 0000.403C.0180.4000.0000.1373  ;3280  D[WORD]_CACHE
U 1373, 0819.0034.8180.09F0.0000.1374  ;3281  ALU_D.AND.K[.3FF],D_ALU,RC[0E]_ALU
U 1374, 0010.0038.61C0.0968.0084.7375  ;3282  Q_RC[0D],SC_K[.F]
U 1375, 0000.003C.0580.0800.0084.B376  ;3283  SC_SC-K[.1]
U 1376, 0001.2034.01C0.09E8.0000.1377  ;3284  Q_Q.AND.MASK,RC[0D]_ALU
U 1377, 001D.0000.0180.0800.0010.1378  ;3285  ALU_D-Q,CLK.UBCC ; ADDRESS CORRECT?
U 1378, 0000.013C.0180.0800.0000.1118  ;3286  Z?
U 1118, 0000.003D.0180.0800.0000.115D  ;3287 =0 ERROR1   ; A-FORK FAILED
U 1119, 0810.0038.0180.0958.0000.1379  ;3288  D_RC[0B]  ; INCREMENT OPCODE ADDRESS
U 1379, 0019.0014.0180.09D8.0000.137A  ;3289  RC[0B]_D+K[.8]
U 137A, 0810.0038.0180.0950.0010.137B  ;3290  D_RC[0A],CLK.UBCC ; ALL OPCODES TESTED?
U 137B, 0000.013C.0180.0800.0000.111A  ;3291  Z?
U 111A, 0019.0000.0580.09D0.0000.1112  ;3292 =0 RC[0A]_D-K[.1],J/AFORK1 ; NO
U 111B, 0000.003C.0180.0800.0000.111C  ;3293  NOP
          ;3294 ;
          ;3295 ; THIS IS A TABLE WHICH CONTAINS OPCODES AND THEIR A-FORK ADDRESSES
          ;3296 ;
          ;3297 ;# 30
          ;3298 ;
          ;3299 ; OPCODE COUNT
          ;3300 ;$ 0026,0000,0000,0000
          ;3301 ;
          ;3302 ; SP MEMORY
          ;3303 ;$ 4040,0051, 01F2,0000
          ;3304 ;
          ;3305 ; SP SHORT LITERAL
          ;3306 ;$ 0F40,0040, 01F6,0000
          ;3307 ;$ 1F40,0040, 01F6,0000
          ;3308 ;$ 2F40,0040, 01F6,0000
          ;3309 ;$ 3F40,0040, 01F6,0000
          ;3310 ;
          ;3311 ; SP LONG LITERAL
          ;3312 ;$ 8F40,4000, 01FA,0000
          ;3313 ;
          ;3314 ; SP REGISTER
          ;3315 ;$ 5140,0040, 01FE,0000
          ;3316 ;
          ;3317 ; DP MEMORY
          ;3318 ;$ 4060,0051, 01F0,0000
          ;3319 ;
          ;3320 ; DP SHORT LITERAL
          ;3321 ;$ 3F60,0040, 01F4,0000
          ;3322 ;
          ;3323 ; DP LONG LITERAL
          ;3324 ;$ 8F60,4000, 01F8,0000
          ;3325 ;$ 4060,0051, 01F0,0000
          ;3326 ;
          ;3327 ; DP REGISTER
          ;3328 ;$ 5160,0040, 01FC,0000
          ;3329 ;
          ;3330 ; SP REGISTER TO REGISTER
```

```
;3331 ;$ 5040,0051, 019E,0000
;3332 ;$ 5341,5152, 01FE,0000
;3333 ;$ 5442,0055, 019E,0000
;3334 ;$ 5743,5556, 01FE,0000
;3335 ;$ 5844,0059, 01AE,0000
;3336 ;$ 5945,5758, 01FE,0000
;3337 ;$ 5A46,005B, 01BE,0000
;3338 ;$ 5947,5B5A, 01FE,0000
;3339 ;$ 50C4,0051, 018E,0000
;3340 ;$ 52C5,5453, 01FE,0000
;3341 ;
;3342 ; SP SHORT LITERAL TO REGISTER
;3343 ;$ 3F40,0050, 0196,0000
;3344 ;$ 3F42,0054, 0196,0000
;3345 ;$ 3F44,0051, 01A6,0000
;3346 ;$ 3F46,0056, 01B6,0000
;3347 ;$ 3FC4,0055, 0186,0000
;3348 ;
;3349 ; DP REGISTER TO REGISTER
;3350 ;$ 5060,0051, 019C,0000
;3351 ;$ 5261,5453, 01FC,0000
;3352 ;$ 5562,0056, 019C,0000
;3353 ;$ 5763,5958, 01FC,0000
;3354 ;$ 5A64,005B, 01AC,0000
;3355 ;$ 5C65,5150, 01FC,0000
;3356 ;$ 5266,0053, 01BC,0000
;3357 ;$ 5467,5655, 01FC,0000
;3358 ;
;3359 ; DP SHORT LITERAL TO REGISTER
;3360 ;$ 3F60,0057, 0194,0000
;3361 ;$ 3F62,0058, 0194,0000
;3362 ;$ 3F64,005A, 01A4,0000
;3363 ;$ 3F66,0055, 01B4,0000
;3364 ;
;3365
```

```
                ;3366  .PAGE "TEST 229 B-FORK TEST"
                ;3367 ;*****************************************************************************
                ;3368 ;**
                ;3369 ; B-FORK TEST
                ;3370 ;
                ;3371 ; TEST DESCRIPTION:
                ;3372 ; THIS IS A TEST OF THE LOGIC IN THE FPA WHICH GENERATES THE B-FORK
                ;3373 ; ADDRESS. THIS TEST MAKES USE OF A TABLE OF OP-CODES IN THE CACHE.
                ;3374 ; PART OF THIS TABLE CONTAINS THE EXPECTED B-FORK ADDRESSES
                ;3375 ; CORRESPONDING TO EACH OPCODE IN THE TABLE.
                ;3376 ;
                ;3377 ; FPA UCODE USED:
                ;3378 ; LOCATION CONTENTS
                ;3379 ; 29:  WAIT,MSC/IR1,NAD/100
                ;3380 ;
                ;3381 ; LOGIC DESCRIPTION:
                ;3382 ;
                ;3383 ; ERROR DESCRIPTION:
                ;3384 ; EXPECTED B-FORK ADDRESS
                ;3385 ; RECEIVED B-FORK ADDRESS
                ;3386 ; OPCODE BEING TESTED
                ;3387 ; ADDRESS OF THAT OPCODE IN THE CACHE
                ;3388 ; LOOP COUNTER
                ;3389 ;
                ;3390 ;--
                ;3391 ;*****************************************************************************
                ;3392 =0
U 111C, 0000.003D.0180.0800.0000.1149  ;3393 BFORK: NEWTST
U 111D, 001B.0010.1180.09F8.0084.737C  ;3394  RC[0F]_0+K[.4]+1,SC_K[.4]
U 137C, 0818.0038.7DF8.0800.0000.137D  ;3395  D_K[.18].Q_0
U 137D, 0D00.003C.0180.0800.0000.137E  ;3396  D_DAL.SC
U 137E, 0001.003C.01E0.0800.0200.137F  ;3397  Q_D,VA_D
U 137F, 0000.003C.0180.C800.0000.1380  ;3398  D[LONG]_CACHE.P  ; GET THE OPCODE COUNT
U 1380, 0001.003C.0180.09D0.0000.1381  ;3399  RC[0A]_D  ; SAVE
U 1381, 0019.2014.0180.09D8.0000.111E  ;3400  RC[0B]_Q+K[.8]  ; SET THE OPCODE ADDRESS POINTER (188)
U 111E, 0000.003D.0180.0800.0000.115C  ;3401 =0 ERLOOP
U 111F, 0000.003C.0180.0800.0000.1120  ;3402  NOP
                ;3403 =0
U 1120, 0000.003D.0180.0800.0000.1166  ;3404 BFORK1: CALL[FPINIT]
U 1121, 0818.0038.2D80.0800.0000.1122  ;3405  D_K[.28]  ; GENERATE ADDRES OF FPA UCODE
U 1122, 0819.0015.0580.0800.0000.116B  ;3406 =0 D_D+K[.1],CALL[GENTRA] ; ...
U 1123, 0001.003C.5980.3E88.0000.1382  ;3407  ID[ACC.2]_D,R[1]_D ; ...
U 1382, 0000.00FC.0380.0800.0000.1383  ;3408  TRAP.FPA  ; TRAP TO THE B-FORK ENTRY
U 1383, 0010.0038.0180.0958.0200.1384  ;3409  VA_RC[0B]  ; GET THE OPCODE
U 1384, 0000.003C.0180.C800.0000.1385  ;3410  D[LONG]_CACHE.P  ; ...
U 1385, 0001.003C.0180.09E0.0000.1124  ;3411  RC[0C]_D  ; SAVE
U 1124, 0000.003D.0180.0800.0000.117B  ;3412 =0 CALL[LOADIB]  ; LOAD THE IB
U 1125, 0000.003C.0180.0800.0000.1040  ;3413  NOP
U 1040, 0000.003D.0180.0800.0000.1041  ;3414 =00 CALL
                ;3415  SUB/SPEC,IBC/CLR.1-5.COND,
U 1041, F000.003F.0180.0800.0000.1200  ;3416       J/IRTAB  ; ISSUE FIRST CALL3
U 1042, 0000.003C.0180.0800.0000.1043  ;3417  NOP
U 1043, 0000.003C.0180.0800.0000.1050  ;3418  NOP
U 1050, 0000.003D.0180.0800.0000.1051  ;3419 =00 CALL
                ;3420  SUB/SPEC,Q_ID[ACC.2],
```

```
U 1051. 0000.003F.59F0.2C00.0000.1200  ;3421       J/IRTAB  ; ISSUE SECOND CALL3
U 1052. 0000.003C.6180.0800.0084.7053  ;3422  SC_K[.F]
U 1053. 0000.003C.0580.0800.0084.B386  ;3423  SC_SC-K[.1]
U 1386. 0001.2034.01C0.09E8.0000.1387  ;3424  Q_Q.AND.MASK,RC[0D]_ALU ; CLEAR BIT 14 OF RECEIVED ACC.2
U 1387. 0810.0038.0180.0958.0000.1388  ;3425  D_RC[0B]  ; READ EXPECTED B-FORK ADDRESS
U 1388. 0019.0014.1180.0800.0200.1389  ;3426  VA_ALU,ALU_D+K[.4] ; ...
U 1389. 0000.003C.0180.C800.0000.138A  ;3427  D[LONG]_CACHE.P  ; ...
U 138A. 081C.2030.0180.0A08.0000.138B  ;3428  D_D.OR.R[1]  ; INSERT TRAP ADDRESS
U 138B. 0000.003C.8D80.0800.0084.738C  ;3429  SC_K[.1F]
U 138C. 0801.0034.0180.09F0.0000.138D  ;3430  D_D.AND.MASK,RC[0E]_ALU ; CLEAR BIT 31
U 138D. 001D.0000.0180.0800.0010.138E  ;3431  ALU_D-Q,CLK.UBCC ; IS ADDRESS CORRECT?
U 138E. 0000.013C.0180.0800.0000.1126  ;3432  Z?
U 1126. 0000.003D.0180.0800.0000.115F  ;3433 =0 ERROR2   ; B-FORK FAILED
U 1127. 0810.0038.0180.0970.0000.138F  ;3434  D_RC[0E]  ; GET EXPECTED DATA AGAIN
U 138F. 0000.003C.59F0.2C00.0000.1390  ;3435  Q_ID[ACC.2]  ; READ ADDRESS AGAIN
U 1390. 0000.003C.6180.0800.0084.7391  ;3436  SC_K[.F]
U 1391. 0000.003C.0580.0800.0084.B392  ;3437  SC_SC-K[.1]
U 1392. 0001.2034.01C0.09E8.0000.1393  ;3438  Q_Q.AND.MASK,RC[0D]_ALU ; CLEAR BIT 14
U 1393. 001D.0000.0180.0800.0010.1394  ;3439  ALU_D-Q,CLK.UBCC ; IS ADDRESS STILL OK?
U 1394. 0000.013C.0180.0800.0000.1128  ;3440  Z?
U 1128. 0000.003D.0180.0800.0000.115F  ;3441 =0 ERROR2   ; B-FORK ADDRESS CHANGED
U 1129. 0810.0038.0180.0958.0000.1395  ;3442  D_RC[0B]  ; INCREMENT OPCODE ADDRESS
U 1395. 0019.0014.0180.09D8.0000.1396  ;3443  RC[0B]_D+K[.8]  ; ...
U 1396. 0810.0038.0180.0950.0010.1397  ;3444  D_RC[0A],CLK.UBCC ; CHECK THE LOOP COUNT
U 1397. 0000.013C.0180.0800.0000.112A  ;3445  Z?
        ;3446 =0 RC[0A]_D-K[.1],
U 112A. 0019.0000.0580.09D0.0000.1060  ;3447       J/CHK_TWINKLE_ECO ; DECREMENT LOOP COUNT
U 112B. 0000.003C.0180.0800.0000.13A0  ;3448  J/BFORK_EXIT  ; TEST IS FINISHED
        ;3449 =00
        ;3450 CHK_TWINKLE_ECO:
U 1060. 0000.003D.0180.0800.0000.1171  ;3451  CALL[CHECK_TWINKLE] ; SEE IF TWINKLE ECO IS INSTALLED
U 1061. 0000.003C.0180.0800.0000.1120  ;3452  J/BFORK1  ; BRANCH IF IT IS
        ;3453   ;
        ;3454   ; TWINKLE ECO IS NOT INSTALLED, CHECK IF LOOP COUNT IS AT
        ;3455   ; TWINKLE PATTERNS.
        ;3456   ;
U 1062. 0810.0038.B980.0950.0000.1398  ;3457  D_RC[0A],K[.19]  ; GET CURRENT LOOP COUNT
U 1398. 0019.0000.B980.0800.0010.1399  ;3458 = ALU_D-K[.19],CLK.UBCC ; AT 19 YET?
U 1399. 0000.013C.0180.0800.0000.112C  ;3459  Z?
U 112C. 0000.003C.6180.0800.0000.139C  ;3460 =0 J/CHK_ECO_2,K[.F] ; NO
        ;3461   ;
        ;3462   ; LOOP COUNT IS AT 19, SKIP 5 PATTERNS
        ;3463   ;
U 112D. 0019.0008.1180.09D0.0000.139A  ;3464  RC[0A]_ALU,ALU_D-K[.4]-1; SKIP 5 PATTERNS
U 139A. 0810.0038.2D80.0958.0000.139B  ;3465  D_RC[0B],K[.28]  ; ...
U 139B. 0019.0014.2D80.09D8.0000.1120  ;3466  RC[0B]_D+K[.28],J/BFORK1; ...
        ;3467   ;
        ;3468   ; IF LOOP COUNT IS AT 0F, THEN SKIP 4 PATTERNS
        ;3469   ;
        ;3470 CHK_ECO_2:
U 139C. 0019.0000.6180.0800.0010.139D  ;3471  ALU_D-K[.F],CLK.UBCC ; IS IT AT F?
U 139D. 0000.013C.0180.0800.0000.112E  ;3472  Z?
U 112E. 0000.003C.0180.0800.0000.1120  ;3473 =0 J/BFORK1  ; NO
U 112F. 0019.0000.1180.09D0.0000.139E  ;3474  RC[0A]_D-K[.4]  ; DECREMENT LOOP COUNT
U 139E. 0810.0038.7580.0958.0000.139F  ;3475  D_RC[0B],K[.20]  ; ...
```

```
U 139F, 0019,0014,7580,09D8,0000,1120  ;3476  RC[0B]_D+K[.20],J/BFORK1; CONTINUE
          ;3477 BFORK_EXIT:
U 13A0, 0000,003C,0180,0800,0000,1130  ;3478  NOP
          ;3479 ;
          ;3480 ; THIS IS A TABLE IN THE CACHE USED IN THE B-FORK TEST. IT CONTAINS
          ;3481 ; OPCODES AND THEIR B-FORK ADDRESSES.
          ;3482 ;
          ;3483 ;* 180
          ;3484 ;
          ;3485 ; OPCODE COUNT
          ;3486 ;$ 0022,0000, 0000,0000
          ;3487 ;
          ;3488 ; SP MEMORY
          ;3489 ;$ 5140,0040, 0132,0000
          ;3490 ;
          ;3491 ; SP SHORT LITERAL
          ;3492 ;$ 5141,500F, 0136,0000
          ;3493 ;$ 5141,501F, 0136,0000
          ;3494 ;$ 5141,502F, 0136,0000
          ;3495 ;$ 5141,503F, 0136,0000
          ;3496 ;
          ;3497 ; SP LONG LITERAL
          ;3498 ;$ 5141,008F, 013A,0000
          ;3499 ;
          ;3500 ; DP MEMORY
          ;3501 ;$ 5160,0040, 0130,0000
          ;3502 ;
          ;3503 ; DP SHORT LITERAL
          ;3504 ;$ 5161,500F, 0134,0000
          ;3505 ;
          ;3506 ; DP LONG LITERAL
          ;3507 ;$ 5161,008F, 0138,0000
          ;3508 ;
          ;3509 ; THE FOLLOWING Rn[Rx] MODES WERE MODIFIED
          ;3510 ; FOR THE TWINKLE ECO. FEBRUARY 1981 *****
          ;3511 ;
          ;3512 ; SP * TO REGISTER
          ;3513 ;$ 4040,0057, 0132,0000 ; **(LC = 19)
          ;3514 ;$ 4442,005D, 0132,0000 ; **
          ;3515 ;$ 4944,0058, 0132,0000 ; **
          ;3516 ;$ 4546,005A, 0132,0000 ; **
          ;3517 ;$ 4CC4,0051, 0132,0000 ; **
          ;3518 ;$ 5741,5A50, 011E,0000
          ;3519 ;$ 5343,555C, 011E,0000
          ;3520 ;$ 5845,5657, 012E,0000
          ;3521 ;$ 5A47,5A55, 013E,0000
          ;3522 ;$ 51C5,515C, 010E,0000
          ;3523 ;
          ;3524 ; DP * TO REGISTER
          ;3525 ;$ 4760,0056, 0130,0000 ; **(LC = 0F)
          ;3526 ;$ 4B62,005A, 0130,0000 ; **
          ;3527 ;$ 4D64,005C, 0130,0000 ; **
          ;3528 ;$ 4C66,0053, 0130,0000 ; **
          ;3529 ;$ 5A61,5859, 011C,0000
          ;3530 ;$ 5A63,505B, 011C,0000
```

```
;3531 ;$ 5C65,5C5D, 012C,0000
;3532 ;$ 5367,535C, 013C,0000
;3533 ;
;3534 ; END OF TWINKLE ECO MODS
;3535 ;
;3536 ; EMODF
;3537 ;$ 5054,0051, 014E,0000
;3538 ;$ 4754,0040, 014E,0000
;3539 ;
;3540 ; EMODD
;3541 ;$ 5B74,0054, 014C,0000
;3542 ;$ 5474,005B, 014C,0000
;3543 ;
;3544 ; POLYF
;3545 ;$ 5055,0051, 015E,0000
;3546 ;$ 4155,0040, 015E,0000
;3547 ;
;3548 ; POLYD
;3549 ;$ 5375,005C, 015C,0000
;3550 ;$ 4C75,0043, 015C,0000
;3551
```

```
            ;3552  .PAGE "TEST 22A FPA ID BUS/ACC.BUS DATA PATH TEST"
            ;3553 ;*************************************************************************
            ;3554 ;+
            ;3555 ; FPA ID BUS/ACC.BUS DATA PATH TEST
            ;3556 ;
            ;3557 ; TEST DESCRIPTION:
            ;3558 ; AN ALL ZEROS/ONES PATTERN IS WRITTEN TO FPA BUS A.
            ;3559 ; THE BUS A DATA IS READ BACK IN THE SAME MICRO STATE.
            ;3560 ;
            ;3561 ; FPA UCODE USED:
            ;3562 ; LOCATION CONTENTS
            ;3563 ;    08F:  BSC/ID,FADC/LD,WAIT,NAD/0FA
            ;3564 ;
            ;3565 ; LOGIC DESCRIPTION:
            ;3566 ; THE FPA ID BUS INTERFACE IS ON THE FML AND FMH MODULES.
            ;3567 ; THE DF MUX (ACCELLATOR BUS) DRIVERS ARE ON THE FNM MODULE.
            ;3568 ;
            ;3569 ; ERROR DESCRIPTION:
            ;3570 ; EXPECTED FPA MICRO ADDRESS
            ;3571 ; RECEIVED FPA MICRO ADDRESS
            ;3572 ; EXPECTED BUS A DATA
            ;3573 ; RECEIVED BUS A DATA
            ;3574 ; LOOP COUNTER
            ;3575 ;
            ;3576 ;-
            ;3577 ;*************************************************************************
            ;3578 =0
U 1130, 0000,003D,0180,0800,0000,1149  ;3579 BTST5: NEWTST
U 1131, 001B,0010,1180,09F8,0000,1132  ;3580  RC[0F]_ALU,ALU_0+K[.4]+1
U 1132, 0000,003D,0180,0800,0000,11B2  ;3581 =0 CALL[TRP.8F]  ; TRAP TO 08F
U 1133, 0018,0038,0980,09D0,0000,13A1  ;3582  RC[0A]_K[.2]  ; SET LOOP COUNTER
U 13A1, 0003,003C,0180,09E0,0000,1134  ;3583  RC[0C]_0  ; EXPECTED DATA
U 1134, 0000,003D,0180,0800,0000,115C  ;3584 =0 ERLOOP
U 1135, 0810,0038,0180,0960,0000,13A2  ;3585 LOOP4: D_RC[0C]  ; GET EXPECTED DATA
U 13A2, 0000,003C,C5D8,3C00,0000,13A3  ;3586  ID[T1]_D,Q_ACC  ; DATA TO/FROM FPA
U 13A3, 0001,203C,0180,09D8,0000,13A4  ;3587  RC[0B]_Q  ; RECEIVED DATA
U 13A4, 081D,2020,0180,0800,0010,13A5  ;3588  Q_Q.XOR.D,CLK.UBCC ; CHECK RESULT
U 13A5, 0000,013C,0180,0800,0000,1136  ;3589  Z?
U 1136, 0000,003D,0180,0800,0000,115D  ;3590 =0 ERROR1
U 1137, 001B,0000,0580,09E0,0000,13A6  ;3591  RC[0C]_ALU,ALU_0-K[.1] ; SET ALL ONES PATTERN
U 13A6, 0810,0038,0180,0950,0000,13A7  ;3592  D_RC[0A]  ; GET LOOP COUNT
U 13A7, 0019,0000,0580,09D0,0010,13A8  ;3593  RC[0A]_D-K[.1],CLK.UBCC ; DECREMENT AND CHECK
U 13A8, 0000,013C,0180,0800,0000,1138  ;3594  Z?
U 1138, 0000,003C,0180,0800,0000,1135  ;3595 =0 J/LOOP4  ; DO ALL ONES PATTERN
U 1139, 0000,003C,0180,0800,0000,113A  ;3596 ENTST4: NOP
            ;3597
```

```
          ;3598  .PAGE "TEST 22B FPA BUS A TEST"
          ;3599 ;*****************************************************************
          ;3600 ;**
          ;3601 ; FPA BUS A TEST
          ;3602 ;
          ;3603 ; TEST DESCRIPTION:
          ;3604 ; A FLOATING ONE/ZERO PATTERN IS WRITTEN TO FPA BUS A. THE TEST
          ;3605 ; PATTERN IS READ BACK IN THE SAME USTATE.
          ;3606 ;
          ;3607 ; FPA UCODE USED:
          ;3608 ; LOCATION CONTENTS
          ;3609 ;   08F:  BSC/ID,FADC/LD,WAIT,NAD/0FA
          ;3610 ;
          ;3611 ; LOGIC DESCRIPTION:
          ;3612 ; THE FPA ID BUS INTERFACE IS ON THE FML AND FMH MODULES.
          ;3613 ; THE DF MUX (ACCELLATOR BUS) DRIVERS ARE ON THE FNM MODULE.
          ;3614 ;
          ;3615 ; ERROR DESCRIPTION:
          ;3616 ; EXPECTED FPA MICRO ADDRESS
          ;3617 ; RECEIVED FPA MICRO ADDRESS
          ;3618 ; EXPECTED BUS A DATA
          ;3619 ; RECEIVED BUS A DATA
          ;3620 ; BIT POSITION COUNTER
          ;3621 ; LOOP COUNTER
          ;3622 ;
          ;3623 ;--
          ;3624 ;*****************************************************************
          ;3625 =0
U 113A. 0000,003D,0180,0800,0000,1149  ;3626 BTST6: NEWTST
U 113B. 0018,0038,D580,09F8,0000,113C  ;3627  RC[0F]_K[.6]
U 113C. 0000,003D,0180,0800,0000,11B2  ;3628 =0 CALL[TRP.8F]  ; TRAP THE FPA
U 113D. 0003,003C,0180,09C8,0000,13A9  ;3629  RC[9]_0   ; PATTERN SELECT FLAG
U 13A9. 0003,003C,1980,09D0,0084,73AA  ;3630  SC_K[ZERO],RC[0A]_0 ; BIT POSITION COUNTER
U 13AA. 0003,001C,0180,09E0,0000,113E  ;3631  RC[0C]_ALU,ALU_NOT.MASK ; INITIAL PATTERN
U 113E. 0000,003D,0180,0800,0000,115C  ;3632 =0 ERLOOP
U 113F. 0810,0038,0180,0960,0000,13AB  ;3633 LOOP5: D_RC[0C]  ; GET TEST PATTERN
U 13AB. 0000,003C,C5D8,3C00,0000,13AC  ;3634  ID[T1]_D,Q_ACC  ; PERFORM THE TEST
U 13AC. 0001,203C,0180,09D8,0000,13AD  ;3635  RC[0B]_Q  ; SAVE RECEIVED DATA
U 13AD. 001D,0020,0180,0800,0010,13AE  ;3636  ALU_D.XOR.Q,CLK.UBCC ; PATTERN OK?
U 13AE. 0000,013C,0180,0800,0000,1140  ;3637  Z?
U 1140. 0000,003D,0180,0800,0000,115F  ;3638 =0 ERROR2   ; A BUS FAILED
U 1141. 0810,0038,0180,0960,0000,13AF  ;3639  D_RC[0C]  ; GET TEST PATTERN
U 13AF. 0010,0038,0180,0948,0010,13B0  ;3640  ALU_RC[9],CLK.UBCC ; WHICH PATTERN ARE WE DOING?
U 13B0. 0000,013C,0180,0800,0000,1142  ;3641  Z?
U 1142. 0000,003C,0180,0800,0000,13B1  ;3642 =0 J/CHKD   ; IT'S A FLOATING ZERO
U 1143. 0000,0D3C,0180,0800,0000,1056  ;3643  D31?   ; FLOATED THE ONE TO BIT 31 YET?
U 1056. 0000,003C,0180,0800,0000,13B5  ;3644 =110 J/RIP1   ; NO
U 1057. 0018,0038,0580,09C8,0000,13B3  ;3645  RC[9]_K[.1],J/RIP0 ; SET PATTERN SELECT FLAG
U 13B1. 0000,193C,0180,0800,0000,106E  ;3646 CHKD: D0?   ; FLOATING ZERO COMPLETE?
U 106E. 0000,003C,0180,0800,0000,13B9  ;3647 =1110 J/ENTST5  ; YES
U 106F. 0810,0038,0180,0950,0000,13B2  ;3648  D_RC[0A]  ; DECREMENT BIT POSITION
U 13B2. 0019,0000,0580,09D0,0082,13B3  ;3649  RC[0A]_D-K[.1],SC_ALU ; ...
U 13B3. 0800,0038,0180,0800,0000,13B4  ;3650 RIP0: D_MASK   ; GET THE NEXT PATTERN
U 13B4. 0001,003C,0180,09E0,0000,113F  ;3651  RC[0C]_D,J/LOOP5 ; CONTINUE TEST
          ;3652
```

```
U 13B5. 0810.0038.0180.0950.0000.13B6  ;3653 RIP1: D_RC[0A]  ; INCREMENT THE BIT POSITION
U 13B6. 0019.0014.0580.09D0.0082.13B7  ;3654  RC[0A]_D+K[.1],SC_ALU ; ...
U 13B7. 0803.001C.0180.0800.0000.13B8  ;3655  D_NOT.MASK  ; GET NEXT PATTERN
U 13B8. 0001.003C.0180.09E0.0000.113F  ;3656  RC[0C]_D,J/LOOP5 ; SAVE AND CONTINUE LOOP
U 13B9. 0000.003C.0180.0800.0000.1144  ;3657 ENTST5: NOP
          ;3658
```

```
            ;3659  .PAGE "TEST 22C RESERVED"
            ;3660 =0
U 1144, 0000,003D,0180,0800,0000,1149  ;3661 T22C: NEWTST
U 1145, 0000,003C,0180,0800,0000,1146  ;3662  NOP
            ;3663
```

```
            ;3664  .PAGE "DUMMY NEWTST"
            ;3665 =0
U 1146. 0000,003D,0180,0800,0000,1149  ;3666 ENDOVR: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2337
ESKARSA.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:12           Page   9
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 1875 2053=
U 1008  2771= 2772= 2773= 2774= 1918= 1919= 1876 2054=
U 1010  2699= 2700= 2701= 2702= 2703= 2704= 2705= 2706=
U 1018  1995= 1996= 1877 2366= 1998= 1999= 1878 2367=
U 1020  3062= 3063= 3070= 3071= 2000= 2001= 2992= 2993=
U 1028  2003= 2004= 1879 2373= 2023= 2024= 1880 2374=
U 1030  3276= 3277= 3278= 3279= 2045= 2046= 2997= 2998=
U 1038  2358= 2359= 1900 2383= 2392= 2393= 1901 2384=
U 1040  3414= 3416= 3417= 3418= 2448= 2449= 3000= 3001=
U 1048  2450= 2451= 1902 2388= 2457= 2458= 1939: 2389=
U 1050  3419= 3421= 3422= 3423= 2487= 2488= 3644= 3645=
U 1058  2489= 2490= 1942: 2401= 2493= 2494= 1945: 2402=
U 1060  3451= 3452= 3457= 1903 2501= 2502= 2506= 2507=
U 1068  2538= 2539= 2542= 2543= 2545= 2546= 3647= 3648=
U 1070  2548= 2551= 2559= 2561= 2562= 2563= 2565= 2568=
U 1078  2576= 2578= 2580= 2581= 2617= 2618= 2622= 2623=
U 1080  2625= 2626= 2627= 2628= 2639= 2640= 2644= 2645=
U 1088  2673= 2674= 2677= 2678= 2680= 2681= 2685= 2686=
U 1090  2691= 2692= 2694= 2695= 2742= 2743= 2745= 2746=
U 1098  2748= 2749= 2750= 2751= 2756= 2758= 2763= 2764=
U 10A0  2766= 2767= 2823= 2824= 2827= 2828= 2830= 2831=
U 10A8  2836= 2837= 2844= 2846= 2847= 2848= 2856= 2857=
U 10B0  2858= 2859= 2863= 2864= 2871= 2872= 2873= 2874=
U 10B8  2875= 2876= 2885= 2887= 2891= 2892= 2926= 2927=
U 10C0  2928= 2929= 2930= 2931= 2944= 2945= 2981= 2982=
U 10C8  2983= 2984= 2988= 2989= 2990= 2991= 2995= 2996=
U 10D0  3003= 3004= 3044= 3045= 3050= 3051= 3053= 3054=
U 10D8  3055= 3056= 3058= 3061= 3072= 3073= 3075= 3076=
U 10E0  3080= 3081= 3083= 3084= 3089= 3090= 3094= 3095=
U 10E8  3097= 3098= 3107= 3111= 3120= 3121= 3127= 3128=
U 10F0  3162= 3163= 3167= 3168= 3170= 3171= 3174= 3175=
U 10F8  3176= 3178= 3184= 3185= 3193= 3194= 3197= 3198=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1904 3259= 3260= 1870: 1871: 1872: 1873:
U 1110  3265= 3266= 3268= 3269= 3270= 3271= 3274= 3275=
U 1118  3287= 3288= 3292= 3293= 3393= 3394= 3401= 3402=
U 1120  3404= 3405= 3406= 3407= 3412= 3413= 3433= 3434=
U 1128  3441= 3442= 3447= 3448= 3460= 3464= 3473= 3474=
U 1130  3579= 3580= 3581= 3582= 3584= 3585= 3590= 3591=
U 1138  3595= 3596= 3626= 3627= 3628= 3629= 3632= 3633=
U 1140  3638= 3639= 3642= 3643= 3661= 3662= 3666= 1905
U 1148  1906 1916 1917 1920 1921 1922 1923 1924
U 1150  1925 1926 1927 1928 1929 1965: 1930 1931
U 1158  1932 1933 1934 1935 1936 1937 1938 1940
U 1160  1941 1973 1974 1980 1981 1982 1993 1994
U 1168  1997 2002 2005 2015 2016 2017 2025 2026
U 1170  2027 2041 2042 2043 2044 2047 2048 2049
U 1178  2050 2051 2052 2061 2062 2063 2064 2065
U 1180  2066 2067 2068 2069 2070 2343 2344 2345
U 1188  2346 2347 2348 2349 2350 2351 2352 2353
U 1190  2354 2355 2356 2357 2362 2363 2364 2365
U 1198  2368 2369 2370 2371 2372 2376 2377 2378
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2338
ESKARSA.MCR                    MICRO2  1P(00)    26-JUL-85  17:13:12     Page   9

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2379 2380 2381 2382 2385 2386 2387 2390
U 11A8   2391 2394 2395 2396 2397 2398 2399 2400
U 11B0   2403 2404 2405 2406 2407 2415 2416 2417
U 11B8   2418 2419 2420 2421 2452 2453 2454 2455
U 11C0   2456 2459 2491 2492 2495 2496 2497 2498
U 11C8   2499 2500 2503 2504 2505 2540 2547 2552
U 11D0   2553 2554 2555 2556 2557 2558 2564 2569
U 11D8   2570 2571 2572 2573 2574 2575 2579 2619
U 11E0   2620 2629 2632 2633 2634 2635 2636 2637
U 11E8   2638 2641 2642 2643 2675 2682 2683 2684
U 11F0   2687 2688 2689 2690 2693 2708 2744 2752
U 11F8   2753 2754 2755 2759 2760 2761 2762 2765
U 1200   2077: 2078: 2079: 2080: 2081: 2082: 2083: 2084:
U 1208   2085: 2086: 2087: 2088: 2089: 2090: 2091: 2092:
U 1210   2093: 2094: 2095: 2096: 2097: 2098: 2099: 2100:
U 1218   2101: 2102: 2103: 2104: 2105: 2106: 2107: 2108:
U 1220   2109: 2110: 2111: 2112: 2113: 2114: 2115: 2116:
U 1228   2117: 2118: 2119: 2120: 2121: 2122: 2123: 2124:
U 1230   2125: 2126: 2127: 2128: 2129: 2130: 2131: 2132:
U 1238   2133: 2134: 2135: 2136: 2137: 2138: 2139: 2140:
U 1240   2141: 2142: 2143: 2144: 2145: 2146: 2147: 2148:
U 1248   2149: 2150: 2151: 2152: 2153: 2154: 2155: 2156:
U 1250   2157: 2158: 2159: 2160: 2161: 2162: 2163: 2164:
U 1258   2165: 2166: 2167: 2168: 2169: 2170: 2171: 2172:
U 1260   2173: 2174: 2175: 2176: 2177: 2178: 2179: 2180:
U 1268   2181: 2182: 2183: 2184: 2185: 2186: 2187: 2188:
U 1270   2189: 2190: 2191: 2192: 2193: 2194: 2195: 2196:
U 1278   2197: 2198: 2199: 2200: 2201: 2202: 2203: 2204:
U 1280   2205: 2206: 2207: 2208: 2209: 2210: 2211: 2212:
U 1288   2213: 2214: 2215: 2216: 2217: 2218: 2219: 2220:
U 1290   2221: 2222: 2223: 2224: 2225: 2226: 2227: 2228:
U 1298   2229: 2230: 2231: 2232: 2233: 2234: 2235: 2236:
U 12A0   2237: 2238: 2239: 2240: 2241: 2242: 2243: 2244:
U 12A8   2245: 2246: 1966: 2247: 2248: 2249: 2250: 2251:
U 12B0   2252: 2253: 2254: 2255: 2256: 2257: 2258: 2259:
U 12B8   2260: 2261: 2262: 2263: 2264: 2265: 2266: 2267:
U 12C0   2268: 2269: 2270: 2271: 2272: 2273: 2274: 2275:
U 12C8   2276: 2277: 2278: 2279: 2280: 2281: 2282: 2283:
U 12D0   2284: 2285: 2286: 2287: 2288: 2289: 2290: 2291:
U 12D8   2292: 2293: 2294: 2295: 2296: 2297: 2298: 2299:
U 12E0   2300: 2301: 2302: 2303: 2304: 2305: 2306: 2307:
U 12E8   2308: 2309: 2310: 2311: 2312: 2313: 2314: 2315:
U 12F0   2316: 2317: 2318: 2319: 2320: 2321: 2322: 2323:
U 12F8   2324: 2325: 2326: 2327: 2328: 2329: 2330: 2331:
U 1300   2776 2825 2826 2832 2833 2834 2835 2838
U 1308   2839 2840 2841 2842 2843 2849 2850 2851
U 1310   2852 2853 2854 2855 2860 2861 2862 2865
U 1318   2866 2867 2868 2869 2870 2877 2878 2879
U 1320   2880 2881 2882 2883 2884 2888 2889 2890
U 1328   2932 2933 2934 2935 2936 2937 2938 2939
U 1330   2940 2941 2942 2943 2985 2986 2987 2994
U 1338   2999 3002 3046 3047 3048 3049 3057 3074
U 1340   3077 3078 3079 3082 3085 3086 3087 3088
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3091 3092 3093 3096 3101 3102 3103 3104
U 1350  3105 3106 3114 3115 3116 3117 3118 3119
U 1358  3125 3126 3164 3165 3172 3173 3179 3180
U 1360  3181 3182 3183 3186 3187 3188 3189 3190
U 1368  3191 3192 3195 3196 3261 3262 3263 3264
U 1370  3272 3273 3280 3281 3282 3283 3284 3285
U 1378  3286 3289 3290 3291 3395 3396 3397 3398
U 1380  3399 3400 3408 3409 3410 3411 3424 3425
U 1388  3426 3427 3428 3429 3430 3431 3432 3435
U 1390  3436 3437 3438 3439 3440 3443 3444 3445
U 1398  3458 3459 3465 3466 3471 3472 3475 3476
U 13A0  3478 3583 3586 3587 3588 3589 3592 3593
U 13A8  3594 3630 3631 3634 3635 3636 3637 3640
U 13B0  3641 3646 3649 3650 3651 3653 3654 3655
U 13B8  3656 3657
U 13C0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR5A	970
Used	970
Remaining	

Total microwords used in memory U: 970
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKARSA.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2342
ESKARSB.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2394 TEST 22D FAD AR, BR LATCH TEST 1
; 2515 TEST 22E FAD AR, BR LATCH TEST 2
; 2760 TEST 22F FPA SCRATCH PAD COPIES DATA TEST 1
; 2847 TEST 230 FPA SCRATCH PADS DATA TEST 2
; 2945 TEST 231 ADDRESSING TEST FOR THE FPA SCRATCH PAD COPIES
; 3023 TEST 232 SP1 ON RA, GENERAL REGISTERS ADDRESSING TEST 1
; 3154 TEST 233 SP2 ON RB, GENERAL REGISTERS ADDRESSING TEST 2
; 3235 TEST 234 SP2+1 ON RA, GENERAL REGISTERS ADDRESSING TEST 3
; 3316 TEST 235 PRN+1 ON RB, GENERAL REGISTERS ADDRESSING TEST 4
; 3398 TEST 236 PRN+1 ON RA, GENERAL REGISTERS ADDRESSING TEST 5
; 3480 TEST 237 RESERVED
; 3485 TEST 238 RESERVED

```
:1  .NOLIST
:1804  .LIST
:1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR5B
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
            ;1841  .PAGE
            ;1842
            ;1843 .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.100A  ;1874 TRPH0: Q_ID[USTACK]
U 100A. 0C00.003C.01E0.0800.0000.100E  ;1875  Q_D,D_Q
U 100E. 0000.003C.8180.3C00.0000.101A  ;1876  ID[USTACK]_D
U 101A. 0C00.003C.01E0.0800.0000.101E  ;1877  Q_D,D_Q
U 101E. 0000.003C.01C0.0A78.0000.102A  ;1878  Q_R[0F]
U 102A. 0001.2024.0180.0800.0010.102E  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102E. 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
                ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1008  ;1897 TRPH1: RC[0D]_SC
U 1008, 0000.003D.D980.0800.0084.7156  ;1898 =0 SETRCF[.9]
U 1009, 0000.003C.49F0.2C00.0000.103A  ;1899  Q_ID[TBER0]
U 103A, 0001.203C.4DF0.2DB0.0000.103E  ;1900  RC[6]_Q,Q_ID[TBER1]
U 103E, 0001.203C.65F0.2DB8.0000.104A  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 104A, 0001.203C.6DF0.2DD8.0000.106A  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 106A, 0001.203C.75F0.2DE0.0000.106E  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 106E, 0001.203C.79F0.2DC8.0000.107A  ;1904  RC[9]_Q,Q_ID[PARITY]
U 107A, 0001.203C.69F0.2DC0.0000.107E  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 107E, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.114F  ;1907 1000: RC[0E]_Q,ERROR1
                ;1908 ;*
                ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1911 ;  NEWTST
                ;1912 ;  ERLOOP
                ;1913 ;  ERROR1
                ;1914 ;  ERROR2
                ;1915 ;
U 108A, 0000.003C.8580.0800.0084.708E  ;1916 SCOPE: SC_K[.C]
U 108E, 0803.001C.0180.0800.0000.100C  ;1917  ALU_NOT.MASK,D_ALU
U 100C, 0000.003D.7580.3C00.0000.1159  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 100D, 0000.003C.65F8.0800.0084.709A  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 109A, 0818.0038.8D80.0800.0000.109E  ;1920  D_K[.1F]    ; ...
U 109E, 0D00.003C.0180.0800.0000.10AA  ;1921  D_DAL.SC
U 10AA, 0000.003C.3D80.3C00.0000.10AE  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 10AE, 0818.0038.0580.0800.0000.10BA  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 10BA, 0D00.003C.0180.0800.0000.10BE  ;1924  D_DAL.SC
U 10BE, 0F00.003C.3180.3C00.0000.10CA  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10CA, 0000.003C.0180.0800.0000.10CE  ;1926  NOP
U 10CE, 0000.003C.3980.3C00.0000.1109  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1109, 0000.003C.2980.3C00.0000.1147  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1147, 0000.003C.4980.3C00.0000.1148  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1148, 0818.0038.C180.0800.0000.1149  ;1930  D_K[.FFFF]
U 1149, 0000.003C.6580.3C00.0000.114A  ;1931  ID[SBI.ERR]_D
U 114A, 0F00.003C.6D80.3C00.0000.114B  ;1932  ID[FAULT]_D,D_0
U 114B, 0000.003C.6D80.3C00.0000.114C  ;1933  ID[FAULT]_D
U 114C, 0000.003C.6580.3C00.0000.114D  ;1934  ID[SBI.ERR]_D
U 114D, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON   ; CALL THE CONSOLE
U 114E, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON   ; ERRLOOP CALL
U 114F, 0810.0038.0180.0978.0000.1150  ;1937 ERROR1: D_RC[0F]
U 1150, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 1151, 0810.0038.0180.0978.0000.1152  ;1940 ERROR2: D_RC[0F]
U 1152, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
                ;1943 105E:
                ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON   ; CALL THE CONSOLE
                ;1946 ;*
                ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
                ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000,003C,0180,0800,0000,13F3  ;1951 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;1952 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;1953 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;1954 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;1955 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;1956 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;1957 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;1958 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;1959 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;1960 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;1961 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;1962 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;1963 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;1964 13FF: NOP
U 1155, 0001,203E,59F0,2DE8,0000,0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001,203E,59F0,2DE8,0000,0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 1153, 0810,0038,4D80,0978,0000,1154  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1154, 0019,4016,4D80,09F8,0000,0001  ;1974  RC[0F]_D*K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 1156, 0010,0038,01C0,0978,0000,1157  ;1980 SETRCF: Q_RC[0F]
U 1157, 0019,2034,4DC0,0800,0000,1158  ;1981  Q_Q.AND.K[.FF00]
U 1158, 0019,2032,1D80,09F8,0000,0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983 ;
            ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;1986 ;   28: NAD/28  ; NOP
            ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;1990 ;  INITIALIZE ITSELF.
            ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;1992 ;
U 1159, 0818,0038,4580,0800,0000,115A  ;1993 FPINIT: D_K[.8000]
U 115A, 0000,003C,5D80,3C00,0000,1018  ;1994  ID[ACC.CS]_D  ; TURN ON FPA
U 1018, 0818,0039,2D80,0800,0000,115E  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1019, 0000,003C,5980,3C00,0000,115B  ;1996  ID[ACC.2]_D
U 115B, 0000,00FC,0380,0800,0000,101C  ;1997  TRAP.FPA  ; TRAP FPA TO 28.
U 101C, 0018,0039,1980,0800,0200,1164  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 101D, 0F00,003C,0180,0800,0000,1028  ;1999  D_0
U 1028, 0000,003D,0180,0800,0000,115E  ;2000 =0 CALL,J/GENTRA
U 1029, 0000,003C,5980,3C00,0000,115C  ;2001  ID[ACC.2]_D
U 115C, 0000,00FC,0380,0800,0000,102C  ;2002  TRAP.FPA  ; TRAP FPA TO 0
U 102C, 0818,0039,2D80,0800,0000,115E  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 102D, 0000,003C,5980,3C00,0000,115D  ;2004  ID[ACC.2]_D
U 115D, 0000,00FE,0380,0800,0000,0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
                ;2006 ;
                ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
                ;2008 ;%0
                ;2009 ;$ 0000.0000, 0000.0000
                ;2010 ;
                ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
                ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
                ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
                ;2014 ;
U 115E. 0000.003C.65F8.0800.0084.715F  ;2015 GENTRA: SC_K[.10],Q_0
U 115F. 0D00.003C.8D80.0800.0084.7160  ;2016  D_DAL.SC,SC_K[.1F]
U 1160. 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
                ;2018 ;
                ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
                ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
                ;2021 ;
                ;2022 =0
U 1034. 0F00.003D.0180.0800.0000.115E  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 1035. 0000.003C.5980.3C00.0000.1161  ;2024  ID[ACC.2]_D
U 1161. 0000.00FC.0380.0800.0000.1162  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 1162. 0000.003C.0180.0800.0000.1163  ;2026  NOP
U 1163. 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
                ;2028 ;
                ;2029 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
                ;2030 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
                ;2031 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
                ;2032 ;
U 1164. 2000.003C.0180.0800.0000.1165  ;2033 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 1165. 3000.003C.0180.6000.0000.1166  ;2034  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 1166. 0000.003C.0180.F800.0000.1167  ;2035 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1167. 0000.003C.0180.F800.0000.1168  ;2036  MCT/ALLOW.IB.READ
U 1168. 0000.003C.0180.F800.0000.1169  ;2037  MCT/ALLOW.IB.READ
U 1169. 0000.003C.0180.F800.0000.116A  ;2038  MCT/ALLOW.IB.READ
U 116A. 0000.003C.0180.F800.0000.116B  ;2039  MCT/ALLOW.IB.READ
U 116B. 1000.003C.0180.F800.0000.116C  ;2040  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 116C. 0000.003E.0180.0800.0000.0001  ;2041  RETURN1
U 116D. 3000.003C.0180.0800.0000.1166  ;2042 WAITIB: IBC/START,J/IBW
                ;2043 ;
                ;2044 ; THIS TABLE IS USED TO CATCH THE IB CALL 3 BRANCH WHEN IT IS
                ;2045 ; NECESSARY FOR A CALL 3 TO BE USED IN FPA TESTING.
                ;2046 ;
                ;2047 1200:
                ;2048 IRTAB:
U 1200. 0001.203E.59F0.2DE8.0000.0002  ;2049 1200: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1201. 0001.203E.59F0.2DE8.0000.0002  ;2050 1201: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1202. 0001.203E.59F0.2DE8.0000.0002  ;2051 1202: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1203. 0001.203E.59F0.2DE8.0000.0002  ;2052 1203: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1204. 0001.203E.59F0.2DE8.0000.0002  ;2053 1204: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1205. 0001.203E.59F0.2DE8.0000.0002  ;2054 1205: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1206. 0001.203E.59F0.2DE8.0000.0002  ;2055 1206: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1207. 0001.203E.59F0.2DE8.0000.0002  ;2056 1207: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1208. 0001.203E.59F0.2DE8.0000.0002  ;2057 1208: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1209. 0001.203E.59F0.2DE8.0000.0002  ;2058 1209: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120A. 0001.203E.59F0.2DE8.0000.0002  ;2059 120A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120B. 0001.203E.59F0.2DE8.0000.0002  ;2060 120B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 120C, 0001.203E.59F0.2DE8.0000.0002  ;2061 120C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 120D, 0001.203E.59F0.2DE8.0000.0002  ;2062 120D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 120E, 0001.203E.59F0.2DE8.0000.0002  ;2063 120E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 120F, 0001.203E.59F0.2DE8.0000.0002  ;2064 120F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1210, 0001.203E.59F0.2DE8.0000.0002  ;2065 1210: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1211, 0001.203E.59F0.2DE8.0000.0002  ;2066 1211: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1212, 0001.203E.59F0.2DE8.0000.0002  ;2067 1212: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1213, 0001.203E.59F0.2DE8.0000.0002  ;2068 1213: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1214, 0001.203E.59F0.2DE8.0000.0002  ;2069 1214: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1215, 0001.203E.59F0.2DE8.0000.0002  ;2070 1215: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1216, 0001.203E.59F0.2DE8.0000.0002  ;2071 1216: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1217, 0001.203E.59F0.2DE8.0000.0002  ;2072 1217: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1218, 0001.203E.59F0.2DE8.0000.0002  ;2073 1218: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1219, 0001.203E.59F0.2DE8.0000.0002  ;2074 1219: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121A, 0001.203E.59F0.2DE8.0000.0002  ;2075 121A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121B, 0001.203E.59F0.2DE8.0000.0002  ;2076 121B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121C, 0001.203E.59F0.2DE8.0000.0002  ;2077 121C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121D, 0001.203E.59F0.2DE8.0000.0002  ;2078 121D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121E, 0001.203E.59F0.2DE8.0000.0002  ;2079 121E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 121F, 0001.203E.59F0.2DE8.0000.0002  ;2080 121F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1220, 0001.203E.59F0.2DE8.0000.0002  ;2081 1220: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1221, 0001.203E.59F0.2DE8.0000.0002  ;2082 1221: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1222, 0001.203E.59F0.2DE8.0000.0002  ;2083 1222: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1223, 0001.203E.59F0.2DE8.0000.0002  ;2084 1223: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1224, 0001.203E.59F0.2DE8.0000.0002  ;2085 1224: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1225, 0001.203E.59F0.2DE8.0000.0002  ;2086 1225: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1226, 0001.203E.59F0.2DE8.0000.0002  ;2087 1226: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1227, 0001.203E.59F0.2DE8.0000.0002  ;2088 1227: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1228, 0001.203E.59F0.2DE8.0000.0002  ;2089 1228: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1229, 0001.203E.59F0.2DE8.0000.0002  ;2090 1229: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122A, 0001.203E.59F0.2DE8.0000.0002  ;2091 122A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122B, 0001.203E.59F0.2DE8.0000.0002  ;2092 122B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122C, 0001.203E.59F0.2DE8.0000.0002  ;2093 122C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122D, 0001.203E.59F0.2DE8.0000.0002  ;2094 122D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122E, 0001.203E.59F0.2DE8.0000.0002  ;2095 122E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 122F, 0001.203E.59F0.2DE8.0000.0002  ;2096 122F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1230, 0001.203E.59F0.2DE8.0000.0002  ;2097 1230: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1231, 0001.203E.59F0.2DE8.0000.0002  ;2098 1231: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1232, 0001.203E.59F0.2DE8.0000.0002  ;2099 1232: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1233, 0001.203E.59F0.2DE8.0000.0002  ;2100 1233: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1234, 0001.203E.59F0.2DE8.0000.0002  ;2101 1234: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1235, 0001.203E.59F0.2DE8.0000.0002  ;2102 1235: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1236, 0001.203E.59F0.2DE8.0000.0002  ;2103 1236: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1237, 0001.203E.59F0.2DE8.0000.0002  ;2104 1237: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1238, 0001.203E.59F0.2DE8.0000.0002  ;2105 1238: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1239, 0001.203E.59F0.2DE8.0000.0002  ;2106 1239: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123A, 0001.203E.59F0.2DE8.0000.0002  ;2107 123A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123B, 0001.203E.59F0.2DE8.0000.0002  ;2108 123B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123C, 0001.203E.59F0.2DE8.0000.0002  ;2109 123C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123D, 0001.203E.59F0.2DE8.0000.0002  ;2110 123D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123E, 0001.203E.59F0.2DE8.0000.0002  ;2111 123E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 123F, 0001.203E.59F0.2DE8.0000.0002  ;2112 123F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1240, 0001.203E.59F0.2DE8.0000.0002  ;2113 1240: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1241, 0001.203E.59F0.2DE8.0000.0002  ;2114 1241: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1242, 0001.203E.59F0.2DE8.0000.0002  ;2115 1242: RC[0D]_Q.Q_ID[ACC.2],RETURN2
```

```
U 1243.  0001.203E.59F0.2DE8.0000.0002  ;2116 1243: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1244.  0001.203E.59F0.2DE8.0000.0002  ;2117 1244: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1245.  0001.203E.59F0.2DE8.0000.0002  ;2118 1245: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1246.  0001.203E.59F0.2DE8.0000.0002  ;2119 1246: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1247.  0001.203E.59F0.2DE8.0000.0002  ;2120 1247: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1248.  0001.203E.59F0.2DE8.0000.0002  ;2121 1248: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1249.  0001.203E.59F0.2DE8.0000.0002  ;2122 1249: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124A.  0001.203E.59F0.2DE8.0000.0002  ;2123 124A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124B.  0001.203E.59F0.2DE8.0000.0002  ;2124 124B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124C.  0001.203E.59F0.2DE8.0000.0002  ;2125 124C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124D.  0001.203E.59F0.2DE8.0000.0002  ;2126 124D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124E.  0001.203E.59F0.2DE8.0000.0002  ;2127 124E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 124F.  0001.203E.59F0.2DE8.0000.0002  ;2128 124F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1250.  0001.203E.59F0.2DE8.0000.0002  ;2129 1250: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1251.  0001.203E.59F0.2DE8.0000.0002  ;2130 1251: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1252.  0001.203E.59F0.2DE8.0000.0002  ;2131 1252: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1253.  0001.203E.59F0.2DE8.0000.0002  ;2132 1253: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1254.  0001.203E.59F0.2DE8.0000.0002  ;2133 1254: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1255.  0001.203E.59F0.2DE8.0000.0002  ;2134 1255: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1256.  0001.203E.59F0.2DE8.0000.0002  ;2135 1256: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1257.  0001.203E.59F0.2DE8.0000.0002  ;2136 1257: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1258.  0001.203E.59F0.2DE8.0000.0002  ;2137 1258: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1259.  0001.203E.59F0.2DE8.0000.0002  ;2138 1259: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125A.  0001.203E.59F0.2DE8.0000.0002  ;2139 125A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125B.  0001.203E.59F0.2DE8.0000.0002  ;2140 125B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125C.  0001.203E.59F0.2DE8.0000.0002  ;2141 125C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125D.  0001.203E.59F0.2DE8.0000.0002  ;2142 125D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125E.  0001.203E.59F0.2DE8.0000.0002  ;2143 125E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 125F.  0001.203E.59F0.2DE8.0000.0002  ;2144 125F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1260.  0001.203E.59F0.2DE8.0000.0002  ;2145 1260: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1261.  0001.203E.59F0.2DE8.0000.0002  ;2146 1261: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1262.  0001.203E.59F0.2DE8.0000.0002  ;2147 1262: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1263.  0001.203E.59F0.2DE8.0000.0002  ;2148 1263: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1264.  0001.203E.59F0.2DE8.0000.0002  ;2149 1264: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1265.  0001.203E.59F0.2DE8.0000.0002  ;2150 1265: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1266.  0001.203E.59F0.2DE8.0000.0002  ;2151 1266: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1267.  0001.203E.59F0.2DE8.0000.0002  ;2152 1267: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1268.  0001.203E.59F0.2DE8.0000.0002  ;2153 1268: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1269.  0001.203E.59F0.2DE8.0000.0002  ;2154 1269: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126A.  0001.203E.59F0.2DE8.0000.0002  ;2155 126A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126B.  0001.203E.59F0.2DE8.0000.0002  ;2156 126B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126C.  0001.203E.59F0.2DE8.0000.0002  ;2157 126C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126D.  0001.203E.59F0.2DE8.0000.0002  ;2158 126D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126E.  0001.203E.59F0.2DE8.0000.0002  ;2159 126E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 126F.  0001.203E.59F0.2DE8.0000.0002  ;2160 126F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1270.  0001.203E.59F0.2DE8.0000.0002  ;2161 1270: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1271.  0001.203E.59F0.2DE8.0000.0002  ;2162 1271: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1272.  0001.203E.59F0.2DE8.0000.0002  ;2163 1272: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1273.  0001.203E.59F0.2DE8.0000.0002  ;2164 1273: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1274.  0001.203E.59F0.2DE8.0000.0002  ;2165 1274: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1275.  0001.203E.59F0.2DE8.0000.0002  ;2166 1275: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1276.  0001.203E.59F0.2DE8.0000.0002  ;2167 1276: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1277.  0001.203E.59F0.2DE8.0000.0002  ;2168 1277: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1278.  0001.203E.59F0.2DE8.0000.0002  ;2169 1278: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1279.  0001.203E.59F0.2DE8.0000.0002  ;2170 1279: RC[0D]_Q.Q_ID[ACC.2].RETURN2
```

```
U 127A. 0001.203E.59F0.2DE8.0000.0002  ;2171 127A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127B. 0001.203E.59F0.2DE8.0000.0002  ;2172 127B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127C. 0001.203E.59F0.2DE8.0000.0002  ;2173 127C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127D. 0001.203E.59F0.2DE8.0000.0002  ;2174 127D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127E. 0001.203E.59F0.2DE8.0000.0002  ;2175 127E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 127F. 0001.203E.59F0.2DE8.0000.0002  ;2176 127F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1280. 0001.203E.59F0.2DE8.0000.0002  ;2177 1280: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1281. 0001.203E.59F0.2DE8.0000.0002  ;2178 1281: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1282. 0001.203E.59F0.2DE8.0000.0002  ;2179 1282: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1283. 0001.203E.59F0.2DE8.0000.0002  ;2180 1283: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1284. 0001.203E.59F0.2DE8.0000.0002  ;2181 1284: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1285. 0001.203E.59F0.2DE8.0000.0002  ;2182 1285: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1286. 0001.203E.59F0.2DE8.0000.0002  ;2183 1286: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1287. 0001.203E.59F0.2DE8.0000.0002  ;2184 1287: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1288. 0001.203E.59F0.2DE8.0000.0002  ;2185 1288: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1289. 0001.203E.59F0.2DE8.0000.0002  ;2186 1289: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128A. 0001.203E.59F0.2DE8.0000.0002  ;2187 128A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128B. 0001.203E.59F0.2DE8.0000.0002  ;2188 128B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128C. 0001.203E.59F0.2DE8.0000.0002  ;2189 128C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128D. 0001.203E.59F0.2DE8.0000.0002  ;2190 128D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128E. 0001.203E.59F0.2DE8.0000.0002  ;2191 128E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 128F. 0001.203E.59F0.2DE8.0000.0002  ;2192 128F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1290. 0001.203E.59F0.2DE8.0000.0002  ;2193 1290: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1291. 0001.203E.59F0.2DE8.0000.0002  ;2194 1291: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1292. 0001.203E.59F0.2DE8.0000.0002  ;2195 1292: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1293. 0001.203E.59F0.2DE8.0000.0002  ;2196 1293: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1294. 0001.203E.59F0.2DE8.0000.0002  ;2197 1294: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1295. 0001.203E.59F0.2DE8.0000.0002  ;2198 1295: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1296. 0001.203E.59F0.2DE8.0000.0002  ;2199 1296: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1297. 0001.203E.59F0.2DE8.0000.0002  ;2200 1297: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1298. 0001.203E.59F0.2DE8.0000.0002  ;2201 1298: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1299. 0001.203E.59F0.2DE8.0000.0002  ;2202 1299: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129A. 0001.203E.59F0.2DE8.0000.0002  ;2203 129A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129B. 0001.203E.59F0.2DE8.0000.0002  ;2204 129B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129C. 0001.203E.59F0.2DE8.0000.0002  ;2205 129C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129D. 0001.203E.59F0.2DE8.0000.0002  ;2206 129D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129E. 0001.203E.59F0.2DE8.0000.0002  ;2207 129E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 129F. 0001.203E.59F0.2DE8.0000.0002  ;2208 129F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A0. 0001.203E.59F0.2DE8.0000.0002  ;2209 12A0: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A1. 0001.203E.59F0.2DE8.0000.0002  ;2210 12A1: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A2. 0001.203E.59F0.2DE8.0000.0002  ;2211 12A2: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A3. 0001.203E.59F0.2DE8.0000.0002  ;2212 12A3: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A4. 0001.203E.59F0.2DE8.0000.0002  ;2213 12A4: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A5. 0001.203E.59F0.2DE8.0000.0002  ;2214 12A5: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A6. 0001.203E.59F0.2DE8.0000.0002  ;2215 12A6: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A7. 0001.203E.59F0.2DE8.0000.0002  ;2216 12A7: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A8. 0001.203E.59F0.2DE8.0000.0002  ;2217 12A8: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12A9. 0001.203E.59F0.2DE8.0000.0002  ;2218 12A9: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12AB. 0001.203E.59F0.2DE8.0000.0002  ;2219 12AB: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12AC. 0001.203E.59F0.2DE8.0000.0002  ;2220 12AC: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12AD. 0001.203E.59F0.2DE8.0000.0002  ;2221 12AD: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12AE. 0001.203E.59F0.2DE8.0000.0002  ;2222 12AE: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12AF. 0001.203E.59F0.2DE8.0000.0002  ;2223 12AF: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B0. 0001.203E.59F0.2DE8.0000.0002  ;2224 12B0: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B1. 0001.203E.59F0.2DE8.0000.0002  ;2225 12B1: RC[0D]_Q.Q_ID[ACC.2].RETURN2
```

```
U 12B2, 0001.203E.59F0.2DE8.0000.0002  ;2226 12B2: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B3, 0001.203E.59F0.2DE8.0000.0002  ;2227 12B3: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B4, 0001.203E.59F0.2DE8.0000.0002  ;2228 12B4: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B5, 0001.203E.59F0.2DE8.0000.0002  ;2229 12B5: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B6, 0001.203E.59F0.2DE8.0000.0002  ;2230 12B6: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B7, 0001.203E.59F0.2DE8.0000.0002  ;2231 12B7: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B8, 0001.203E.59F0.2DE8.0000.0002  ;2232 12B8: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12B9, 0001.203E.59F0.2DE8.0000.0002  ;2233 12B9: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BA, 0001.203E.59F0.2DE8.0000.0002  ;2234 12BA: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BB, 0001.203E.59F0.2DE8.0000.0002  ;2235 12BB: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BC, 0001.203E.59F0.2DE8.0000.0002  ;2236 12BC: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BD, 0001.203E.59F0.2DE8.0000.0002  ;2237 12BD: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BE, 0001.203E.59F0.2DE8.0000.0002  ;2238 12BE: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12BF, 0001.203E.59F0.2DE8.0000.0002  ;2239 12BF: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C0, 0001.203E.59F0.2DE8.0000.0002  ;2240 12C0: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C1, 0001.203E.59F0.2DE8.0000.0002  ;2241 12C1: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C2, 0001.203E.59F0.2DE8.0000.0002  ;2242 12C2: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C3, 0001.203E.59F0.2DE8.0000.0002  ;2243 12C3: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C4, 0001.203E.59F0.2DE8.0000.0002  ;2244 12C4: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C5, 0001.203E.59F0.2DE8.0000.0002  ;2245 12C5: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C6, 0001.203E.59F0.2DE8.0000.0002  ;2246 12C6: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C7, 0001.203E.59F0.2DE8.0000.0002  ;2247 12C7: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C8, 0001.203E.59F0.2DE8.0000.0002  ;2248 12C8: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12C9, 0001.203E.59F0.2DE8.0000.0002  ;2249 12C9: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CA, 0001.203E.59F0.2DE8.0000.0002  ;2250 12CA: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CB, 0001.203E.59F0.2DE8.0000.0002  ;2251 12CB: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CC, 0001.203E.59F0.2DE8.0000.0002  ;2252 12CC: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CD, 0001.203E.59F0.2DE8.0000.0002  ;2253 12CD: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CE, 0001.203E.59F0.2DE8.0000.0002  ;2254 12CE: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12CF, 0001.203E.59F0.2DE8.0000.0002  ;2255 12CF: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D0, 0001.203E.59F0.2DE8.0000.0002  ;2256 12D0: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D1, 0001.203E.59F0.2DE8.0000.0002  ;2257 12D1: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D2, 0001.203E.59F0.2DE8.0000.0002  ;2258 12D2: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D3, 0001.203E.59F0.2DE8.0000.0002  ;2259 12D3: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D4, 0001.203E.59F0.2DE8.0000.0002  ;2260 12D4: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D5, 0001.203E.59F0.2DE8.0000.0002  ;2261 12D5: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D6, 0001.203E.59F0.2DE8.0000.0002  ;2262 12D6: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D7, 0001.203E.59F0.2DE8.0000.0002  ;2263 12D7: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D8, 0001.203E.59F0.2DE8.0000.0002  ;2264 12D8: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12D9, 0001.203E.59F0.2DE8.0000.0002  ;2265 12D9: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DA, 0001.203E.59F0.2DE8.0000.0002  ;2266 12DA: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DB, 0001.203E.59F0.2DE8.0000.0002  ;2267 12DB: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DC, 0001.203E.59F0.2DE8.0000.0002  ;2268 12DC: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DD, 0001.203E.59F0.2DE8.0000.0002  ;2269 12DD: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DE, 0001.203E.59F0.2DE8.0000.0002  ;2270 12DE: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12DF, 0001.203E.59F0.2DE8.0000.0002  ;2271 12DF: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E0, 0001.203E.59F0.2DE8.0000.0002  ;2272 12E0: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E1, 0001.203E.59F0.2DE8.0000.0002  ;2273 12E1: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E2, 0001.203E.59F0.2DE8.0000.0002  ;2274 12E2: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E3, 0001.203E.59F0.2DE8.0000.0002  ;2275 12E3: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E4, 0001.203E.59F0.2DE8.0000.0002  ;2276 12E4: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E5, 0001.203E.59F0.2DE8.0000.0002  ;2277 12E5: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E6, 0001.203E.59F0.2DE8.0000.0002  ;2278 12E6: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E7, 0001.203E.59F0.2DE8.0000.0002  ;2279 12E7: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 12E8, 0001.203E.59F0.2DE8.0000.0002  ;2280 12E8: RC[0D]_Q.Q_ID[ACC.2].RETURN2
```

```
U 12E9. 0001.203E.59F0.2DE8.0000.0002  ;2281 12E9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EA. 0001.203E.59F0.2DE8.0000.0002  ;2282 12EA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EB. 0001.203E.59F0.2DE8.0000.0002  ;2283 12EB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EC. 0001.203E.59F0.2DE8.0000.0002  ;2284 12EC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12ED. 0001.203E.59F0.2DE8.0000.0002  ;2285 12ED: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EE. 0001.203E.59F0.2DE8.0000.0002  ;2286 12EE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EF. 0001.203E.59F0.2DE8.0000.0002  ;2287 12EF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F0. 0001.203E.59F0.2DE8.0000.0002  ;2288 12F0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F1. 0001.203E.59F0.2DE8.0000.0002  ;2289 12F1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F2. 0001.203E.59F0.2DE8.0000.0002  ;2290 12F2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F3. 0001.203E.59F0.2DE8.0000.0002  ;2291 12F3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F4. 0001.203E.59F0.2DE8.0000.0002  ;2292 12F4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F5. 0001.203E.59F0.2DE8.0000.0002  ;2293 12F5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F6. 0001.203E.59F0.2DE8.0000.0002  ;2294 12F6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F7. 0001.203E.59F0.2DE8.0000.0002  ;2295 12F7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F8. 0001.203E.59F0.2DE8.0000.0002  ;2296 12F8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F9. 0001.203E.59F0.2DE8.0000.0002  ;2297 12F9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FA. 0001.203E.59F0.2DE8.0000.0002  ;2298 12FA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FB. 0001.203E.59F0.2DE8.0000.0002  ;2299 12FB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FC. 0001.203E.59F0.2DE8.0000.0002  ;2300 12FC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FD. 0001.203E.59F0.2DE8.0000.0002  ;2301 12FD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FE. 0001.203E.59F0.2DE8.0000.0002  ;2302 12FE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FF. 0001.203E.59F0.2DE8.0000.0002  ;2303 12FF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;2304 ;
            ;2305 ; THIS SUBROUTINE IS USED IN THE FPA SCRATCH PAD TEST.
            ;2306 ; IT SETS UP UNIQUE DATA PATTERNS IN EACH REGISTER SO THAT
            ;2307 ; ADDRESSING TESTS CAN BE PERFORMED. THE DATA GENERATED IS:
            ;2308 ; R[0]  00000000
            ;2309 ; R[1]  11111111
            ;2310 ; R[2]  22222222
            ;2311 ;  ...
            ;2312 ; R[0E]  EEEEEEEE
            ;2313 ; R[0F]  FFFFFFFF
            ;2314 ;
U 116E. 0018.0038.61C0.0800.0000.116F  ;2315 RABTAB: Q_K[.F]
U 116F. 0000.003C.ED80.0800.0084.7170  ;2316 RABP1: SC_K[.1B]
U 1170. 0C00.003C.05F8.0800.0084.9171  ;2317  SC_SC+K[.1],D_Q,Q_0
U 1171. 0D00.003C.0180.0800.0000.1172  ;2318  D_DAL.SC
U 1172. 0F00.003C.11E0.0800.0084.7173  ;2319  Q_D,SC_K[.4],D_0
U 1173. 0D00.003C.0180.0800.0000.1174  ;2320  D_DAL.SC
U 1174. 0D00.003C.0180.0800.0000.1175  ;2321  D_DAL.SC
U 1175. 0D00.003C.0180.0800.0000.1176  ;2322  D_DAL.SC
U 1176. 0D00.003C.0180.0800.0000.1177  ;2323  D_DAL.SC
U 1177. 0D00.003C.0180.0800.0000.1178  ;2324  D_DAL.SC
U 1178. 0D00.003C.0180.0800.0000.1179  ;2325  D_DAL.SC
U 1179. 0D00.003C.0180.0800.0000.117A  ;2326  D_DAL.SC
U 117A. 0D00.003C.0180.0800.0000.117B  ;2327  D_DAL.SC
U 117B. 0019.0034.61C0.0800.0092.117C  ;2328  ALU_D[AND]K[.F],SC_ALU,Q_ALU,CLK.UBCC
U 117C. 0001.013C.0180.08E8.0000.1036  ;2329  Z?,ALU_D,R(SC)_ALU
U 1036. 0019.2000.05C0.0800.0000.116F  ;2330 =0 ALU_Q-K[.1],Q_ALU,J/RABP1
U 1037. 0000.003E.0180.0800.0000.0001  ;2331  RETURN1
            ;2332 ; SHIFT D REGISTTER CONTENTS
            ;2333 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
U 117D. 0000.003C.01F8.0800.0000.117E  ;2334 SHIFTD: Q_0
U 117E. 0500.003C.0180.0800.0000.117F  ;2335  D_D.LEFT
```

```
U 117F. 0000.003C.0580.0800.1414.B180  ;2336  EALU_STATE-K[.1],CLK.UBCC
U 1180. 0000.123C.0180.0800.0000.100B  ;2337  EALU.Z?
U 100B. 0000.003C.0180.0800.0000.117D  ;2338 =1011 J/SHIFTD
U 100F. 0000.003E.0180.0800.0000.0001  ;2339  RETURN1
U 1181. 0000.003C.01F8.0800.0000.1182  ;2340 SHIFQD: Q_0
U 1182. 0001.2028.01C0.0800.0000.1183  ;2341  Q_NOT.Q
U 1183. 0500.003C.0280.0800.0000.1184  ;2342  D_D.LEFT.Q  ; FLOATING ZERO PATTERN
U 1184. 0000.003C.0580.0800.1414.B185  ;2343  EALU_STATE-K[.1],CLK.UBCC
U 1185. 0000.123C.0180.0800.0000.101B  ;2344  EALU.Z?
U 101B. 0000.003C.0180.0800.0000.1181  ;2345 =1011 J/SHIFQD
U 101F. 0000.003E.0180.0800.0000.0001  ;2346  RETURN1
          ;2347 ; THIS ROUTINE LOADS EACH NIBBLE OF THE D REGISTER WITH THE PATTERN HELD BY THE SC
U 1186. 0000.003C.01F8.0800.0000.1187  ;2348 SHIFD4: Q_0
U 1187. 0000.003C.0180.0800.1404.7188  ;2349  STATE_K[.8]  ;
U 1188. 0100.003C.0180.0800.0000.1189  ;2350 SHIFT: D_D.LEFT2
U 1189. 0100.003C.0180.0800.0000.118A  ;2351  D_D.LEFT2
U 118A. 0811.0030.0180.0948.0000.118B  ;2352  D_D.OR.RC[9]
U 118B. 0000.003C.0580.0800.1414.B18C  ;2353  EALU_STATE-K[.1],CLK.UBCC
U 118C. 0000.123C.0180.0800.0000.102B  ;2354  EALU.Z?
U 102B. 0000.003C.0180.0800.0000.1188  ;2355 =1011 J/SHIFT
U 102F. 0000.003E.0180.0800.0000.0001  ;2356  RETURN1
U 118D. 0000.003C.0180.0800.0080.D18E  ;2357 SC.CHK: SC_SC+1
U 118E. 0000.003C.6580.0800.0014.B18F  ;2358  EALU_SC-K[.10],CLK.UBCC
U 118F. 0000.123C.0180.0800.0000.103B  ;2359  EALU.Z?
U 103B. 0000.003E.0180.0800.0000.0001  ;2360 =1011 RETURN1
U 103F. 0000.003E.0180.0800.0000.0002  ;2361  RETURN2
U 1190. 0819.0030.4580.0800.0000.1191  ;2362 TRAP: D_D.OR.K[.8000]  ; TRAP ENABLE BIT
U 1191. 0000.003C.6580.0800.1404.7038  ;2363  STATE_K[.10]  ; BIT 16
U 1038. 0000.003D.0180.0800.0000.117D  ;2364 =0 CALL[SHIFTD]
U 1039. 0001.003C.0180.09F0.0000.1192  ;2365  RC[0E]_D  ; EXPECTED ADDRESS
U 1192. 0000.003C.5980.3C00.0000.1193  ;2366  ID[ACC.2]_D  ; WRITE TRAP ADDRESS
U 1193. 0000.00FC.0380.0800.0000.1194  ;2367  TRAP.FPA  ; TRAP DIRECTIVE
U 1194. 0000.003C.59F0.2C00.0000.1195  ;2368  Q_ID[ACC.2]  ; READ FPA MAINTENANCE REGISTER
U 1195. 0001.203E.0180.09E8.0000.0001  ;2369  RC[0D]_Q,RETURN1 ; SAVE AND RETURN
U 1196. 0000.007C.0180.0800.0000.1197  ;2370 STEP: CPSYNC
U 1197. 0000.003C.0580.0800.1414.B198  ;2371  EALU_STATE-K[.1],CLK.UBCC
U 1198. 0000.123C.0180.0800.0000.104B  ;2372  EALU.Z?
U 104B. 0000.003C.0180.0800.0000.1196  ;2373 =1011 J/STEP
U 104F. 0000.003E.0180.0800.0000.0001  ;2374  RETURN1
U 1199. 0818.0038.4180.0800.0000.119A  ;2375 TRP.81: D_K[.80]
U 119A. 0819.0014.0580.0800.0000.1190  ;2376  D_D+K[.1],J/TRAP
U 119B. 0818.0038.4180.0800.0000.119C  ;2377 TRP 8F: D_K[.80]
U 119C. 0000.003C.6180.0800.0000.119D  ;2378  KM./.F
U 119D. 0819.0014.6180.0800.0000.1190  ;2379  D_D+K[.F],J/TRAP
          ;2380 ;*
          ;2381 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
          ;2382 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE IB
          ;2383 ; IS COMPLETELY FULL OF DATA.
          ;2384 ;
          ;2385 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
          ;2386 ;-
U 119E. 3000.003C.0180.6000.0000.119F  ;2387 IBLOAD: LOAD.IB,IBC/START
U 119F. 0000.003C.0180.F800.0000.11A0  ;2388 SYNC02: MCT/ALLOW.IB.READ
U 11A0. 0000.003C.0180.F800.0000.11A1  ;2389  MCT/ALLOW.IB.READ
U 11A1. 0000.003C.0180.F800.0000.11A2  ;2390  MCT/ALLOW.IB.READ
```

```
U 11A2, 0000,003C,0180,F800,0000,11A3  ;2391  MCT/ALLOW.IB.READ
U 11A3, 0000,003C,0180,F800,0000,11A4  ;2392  MCT/ALLOW.IB.READ
U 11A4, 1000,003E,0180,F800,0000,0001  ;2393  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
```

```
;2394   .PAGE "TEST 22D FAD AR, BR LATCH TEST 1"
;2395  ;*****************************************************************
;2396  ;+*
;2397  ;  FAD AR, BR LATCH TEST 1
;2398  ;
;2399  ;
;2400  ; TEST DESCRIPTION AN ALL ZEROS/ONES TEST PATTERN IS USED AS
;2401  ;   DESCRIBEDIN THE FOLLOWING TEST SEQUENCE.
;2402  ;
;2403  ; TEST SEQUENCE  DESCRIPTION
;2404  ; (1)  GENERATE TEST PATTERNS
;2405  ; (2)  LOAD AR, BR LATCHES. AR_ZEROS, BR_ONES
;2406  ; (3)  READ BACK AR HI, AR LO, BR HI, BR LO
;2407  ; (4)  VERIFY RESULTS
;2408  ; (5)  REVERSE TEST PATTERN IN AR,BR AND
;2409  ;   REPEAT TEST SEQUENCE STEPS 3,4,5.
;2410  ;
;2411  ; FPA UCODE USED
;2412  ; LOCATION CONTENTS
;2413  ; 08F: BSC/ID,FADC/LD,WAIT,NAD/0FA
;2414  ; 0FA: BSC/ID,FADC/AR,WAIT,FPSYNC,NAD/0FB
;2415  ; 0FB: BSC/FAL.HL,FADC/A,WAIT,NAD/0FC
;2416  ; 0FC: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/0FD
;2417  ; 0FD: BSC/FAL.HL,FADC/B,WAIT,FPSYNC,NAD/0FE
;2418  ; 0FE: BSC/FAL.LH,FADC/B,WAIT,FPSYNC,NAD/0FE
;2419  ;
;2420  ; LOGIC DESCRIPTION
;2421  ; THE FAD ADDER MUX ENABLE AND SELECT LOGIC, AND
;2422  ; THE FALU FUNCTION SELECT LOGIC IS ON THE
;2423  ; FCT MODULE.
;2424  ; THE AR LATCH,ADDER MUX, FALU, AND FALU MUX ARE
;2425  ; ON THE FAD MODULE.
;2426  ;
;2427  ; ERROR DESCRIPTION
;2428  ; EXPECTED TRAP UADDRESS
;2429  ; RECEIVED TRAP UADDRESS
;2430  ; EXPECTED AR HI
;2431  ; RECEIVED AR HI
;2432  ; EXPECTED AR LO
;2433  ; RECEIVED AR LO
;2434  ; EXPECTED BR HI
;2435  ; RECEIVED BR HI
;2436  ; EXPECTED BR LO
;2437  ; RECEIVED BR LO
;2438  ; MASK FOR BITS <15,7>
;2439  ; MASK FOR BITS <15:7>
;2440  ; LOOP COUNTER
;2441  ; ERROR1 COUNTER
;2442  ;
;2443  ; SYNC POINT DESCRIPTION
;2444  ;--
;2445  ;*****************************************************************
;2446  ;/////////////////////////////////////////
;2447  ;+*
;2448  ; TEST FAD AR BR LATCHES WITH ALL ZEROS/ONES PATTERN
```

```
                ;2449 ;-
                ;2450 =0
U 103C. 0018.0039.F580.09F8.0000.108A  ;2451 BTST7: NEWTST[.A]
U 103D. 0810.0038.0180.0978.0000.11A5  ;2452  D_RC[0F]
U 11A5. 0019.0014.1180.09F8.0000.11A6  ;2453  RC[0F]_D+K[.4]  ; 14 ARGUMENTS
U 11A6. 0818.0038.7580.0800.0000.1040  ;2454  D_K[.20]
U 1040. 2019.0015.1180.0800.4200.119E  ;2455 =0 ALU_D+K[.4],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 1041. 0018.0038.0980.0990.0000.11A7  ;2456  RC[2]_K[.2] ; LOOP COUNTER
U 11A7. 0818.0038.4D80.0800.0000.11A8  ;2457  D_K[.FF00]
U 11A8. 0819.0030.4180.0800.0000.11A9  ;2458  D_D.OR.K[.80] ; MASK OUT SIGN AND EXPONENT <15:07>
U 11A9. 0001.0028.0180.0998.0000.11AA  ;2459  RC[3]_NOT.D ; SAVE MASK FOR BR HI
U 11AA. 0818.0038.4580.0800.0000.11AB  ;2460  D_K[.8000]
U 11AB. 0819.0030.4180.0800.0000.11AC  ;2461  D_D.OR.K[.80]
U 11AC. 0001.0028.0180.09A0.0000.11AD  ;2462  RC[4]_NOT.D ; MASK FOR AR HI (PASS ARX BITS)
U 11AD. 0F00.003C.0180.0800.0000.11AE  ;2463  D_0
U 11AE. 0001.003C.0180.09D0.0000.11AF  ;2464  RC[0A]_D ; EXPECTED AR  LO
U 11AF. 0801.0028.0180.0800.0000.11B0  ;2465  D_NOT.D
U 11B0. 0001.003C.0180.09B0.0000.1042  ;2466  RC[6]_D  ; EXPECTED BR LO
U 1042. 0000.003D.0180.0800.0000.114E  ;2467 =0 ERLOOP
U 1043. 0000.003C.0180.0800.0000.1044  ;2468  NOP
                ;2469 =0
U 1044. 0000.003D.0180.0800.0000.119B  ;2470 LOOP6: CALL[TRP.8F] ; TRAP TO ADDRESS 08F
U 1045. 0810.0038.0180.0930.0000.11B1  ;2471  D_RC[6]  ; DATA AR ,BR
U 11B1. 0000.007C.C580.3C00.0000.11B2  ;2472  ID[T1]_D,CPSYNC ; LOAD AR, BR, THEN  GO  TO ADDRESS 0FA
U 11B2. 0810.0038.0180.0950.0000.11B3  ;2473  D_RC[0A] ; DATA AR  HI  LO
U 11B3. 0000.007C.C580.3C00.0000.11B4  ;2474  ID[T1]_D,CPSYNC ; LOAD AR HI, LO, THEN  GO  TO ADDRESS 0FB
U 11B4. 0000.007C.01D8.0800.0000.11B5  ;2475  Q_ACC,CPSYNC ; READ AR HI,THEN GO TO  0FC
U 11B5. 0011.2034.01C0.0920.0000.11B6  ;2476  Q_Q.AND.RC[4]  ; MASK OUT BITS 15,7
U 11B6. 0001.203C.0180.09D8.0000.11B7  ;2477  RC[0B]_Q ; SAVE
U 11B7. 0000.007C.01D8.0800.0000.11B8  ;2478  Q_ACC,CPSYNC ; READ AR LO, THEN GO TO  0FD
U 11B8. 0001.203C.0180.09C8.0000.11B9  ;2479  RC[9]_Q  ; SAVE
U 11B9. 0000.007C.01D8.0800.0000.11BA  ;2480  Q_ACC,CPSYNC ; READ BR HI,THEN GO TO  0FE
U 11BA. 0011.2034.01C0.0918.0000.11BB  ;2481  Q_Q.AND.RC[3] ; MASK OUT <15:07>
U 11BB. 0001.203C.0180.09B8.0000.11BC  ;2482  RC[7]_Q  ; SAVE
U 11BC. 0000.003C.01D8.0800.0000.11BD  ;2483  Q_ACC  ; READ BR LO
U 11BD. 0001.203C.0180.09A8.0000.11BE  ;2484  RC[5]_Q  ; SAVE
U 11BE. 0010.0038.01C0.0958.0000.11BF  ;2485  Q_RC[0B] ; CHECK AR HI
U 11BF. 0810.0038.0180.0950.0000.11C0  ;2486  D_RC[0A]
U 11C0. 0811.0034.0180.0920.0000.11C1  ;2487  D_D.AND.RC[4] ; MASK OUT BITS 15,7
U 11C1. 0001.003C.0180.09E0.0000.11C2  ;2488  RC[0C]_D ; EXPECTED AR HI
U 11C2. 0011.2020.0180.0960.0010.11C3  ;2489  ALU_Q.XOR.RC[0C],CLK.UBCC
U 11C3. 0000.013C.0180.0800.0000.1046  ;2490  Z?
U 1046. 0018.0039.0580.0988.0000.114F  ;2491 =0 ERROR1,RC[1]_K[.1]
U 1047. 0010.0038.01C0.0948.0000.11C4  ;2492  Q_RC[9]  ; CHECK AR LO
U 11C4. 0011.2020.0180.0950.0010.11C5  ;2493  ALU_Q.XOR.RC[0A],CLK.UBCC
U 11C5. 0000.013C.0180.0800.0000.1048  ;2494  Z?
U 1048. 0018.0039.0980.0988.0000.114F  ;2495 =0 ERROR1,RC[1]_K[.2]
U 1049. 0810.0038.0180.0930.0000.11C6  ;2496  D_RC[6]  ; RESTORE D REGISTER
U 11C6. 0010.0038.01C0.0938.0000.11C7  ;2497  Q_RC[7]  ; CHECK BR HI
U 11C7. 0811.0034.0180.0918.0000.11C8  ;2498  D_D.AND.RC[3]
U 11C8. 0001.003C.0180.09C0.0000.11C9  ;2499  RC[8]_D  ; EXPECTED BR HI
U 11C9. 0011.2020.0180.0940.0010.11CA  ;2500  ALU_Q.XOR.RC[8],CLK.UBCC
U 11CA. 0000.013C.0180.0800.0000.104C  ;2501  Z?
U 104C. 0018.0039.0D80.0988.0000.114F  ;2502 =0 ERROR1,RC[1]_K[.3]
U 104D. 0010.0038.01C0.0928.0000.11CB  ;2503  Q_RC[5]  ; CHECK BR LO
```

```
U 11CB. 0011.2020.0180.0930.0010.11CC  ;2504  ALU_Q.XOR.RC[6],CLK.UBCC
U 11CC. 0000.013C.0180.0800.0000.1050  ;2505  Z?
U 1050. 0018.0039.1180.0988.0000.114F  ;2506 =0 ERROR1,RC[1]_K[.4]
U 1051. 0003.003C.0180.09B0.0000.11CD  ;2507  RC[6]_0   ; BR  TO ZEROS
U 11CD. 0003.0028.0180.09E0.0000.11CE  ;2508  RC[0C]_NOT.0 ; AR  TO ONES
U 11CE. 0810.0038.0180.0910.0000.11CF  ;2509  D_RC[2]   ; CHECK LOOP COUNT
U 11CF. 0019.0000.0580.0990.0010.11D0  ;2510  RC[2]_D-K[.1],CLK.UBCC
U 11D0. 0000.013C.0180.0800.0000.1052  ;2511  Z?
U 1052. 0000.003C.0180.0800.0000.1044  ;2512 =0 J/LOOP6
U 1053. 0000.003C.0180.0800.0000.1054  ;2513 ENTST6: NOP
           ;2514
```

```
;2515  .PAGE "TEST 22E FAD AR, BR LATCH TEST 2"
;2516 ;*****************************************************************
;2517 ;**
;2518 ; FAD AR, BR LATCH TEST 2
;2519 ;
;2520 ; TEST DESCRIPTION A FLOATING ONE/ZERO TEST PATTERN IS USED AS
;2521 ;   DESCRIBED IN THE FOLLOWING TEST SEQUENCE.
;2522 ;
;2523 ; TEST SEQUENCE  DESCRIPTION
;2524 ; (1)  GENERATE TEST PATTERN
;2525 ; (2)  WRITE PATTERN INTO THE AR, BR LATCHES
;2526 ; (3)  READ BACK AR, BR LATCHES
;2527 ; (4)  VERIFY RESULTS
;2528 ; (5)  REPEAT STEPS 2,3,4 UNTILL ALL BITS
;2529 ;   IN THE AR, BR LATCHES HAVE BEEN
;2530 ;   VERIFIED.
;2531 ;
;2532 ; FPA UCODE USED
;2533 ; LOCATION CONTENTS
;2534 ; 08F: BSC/ID,FADC/LD,WAIT,NAD/0FA
;2535 ; 0FA: BSC/ID,FADC/AR,WAIT,FPSYNC,NAD/0FB
;2536 ; 0FB: BSC/FAL.HL,FADC/A,WAIT,NAD/0FC
;2537 ; 0FC: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/0FD
;2538 ; 0FD: BSC/FAL.HL,FADC/B,WAIT,FPSYNC,NAD/0FE
;2539 ; 0FE: BSC/FAL.LH,FADC/B,WAIT,FPSYNC,NAD/0FE
;2540 ;
;2541 ; LOGIC DESCRIPTION
;2542 ; THE FAD ADDER MUX ENABLE AND SELECT LOGIC, AND
;2543 ; THE FALU FUNCTION SELECT LOGIC IS ON THE
;2544 ; FCT MODULE.
;2545 ; THE AR LATCH,ADDER MUX, FALU, AND FALU MUX ARE
;2546 ; ON THE FAD MODULE.
;2547 ;
;2548 ; ERROR DESCRIPTION
;2549 ; EXPECTED TRAP UADDRESS
;2550 ; RECEIVED TRAP UADDRESS
;2551 ; EXPECTED AR HI
;2552 ; RECEIVED AR HI
;2553 ; EXPECTED AR LO
;2554 ; RECEIVED AR LO
;2555 ; MASK FOR BITS <15,7>
;2556 ; BIT POSITION COUNTER
;2557 ;
;2558 ; SYNC POINT DESCRIPTION
;2559 ;--
;2560 ;*****************************************************************************
;2561 ;/////////////////////////////////////////////
;2562 ;*
;2563 ; SUBTEST 1
;2564 ; TEST AR LATCH WITH A FLOATING ONE PATTERN
;2565 ;-
;2566 =0
U 1054, 0018,0039,1180,09F8,0000,108A  ;2567 BTST8: NEWTST[.4]
U 1055, 0810,0038,0180,0978,0000,11D1  ;2568  D_RC[0F]
U 11D1, 0019,0014,1180,09F8,0000,1056  ;2569  RC[0F]_D+K[.4]  ; 8 ARGUMENTS
```

```
U 1056, 2018,0039,0180,0800,4200,119E  ;2570 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 1057, 0818,0038,4580,0800,0000,11D2  ;2571  D_K[.8000]
U 11D2, 0819,0030,4180,0800,0000,11D3  ;2572  D_D.OR.K[.80]
U 11D3, 0001,0028,0180,09C0,0000,11D4  ;2573  RC[8]_NOT.D   ; SAVE MASK FOR AR HI
U 11D4, 0818,0038,0580,0800,0000,11D5  ;2574  D_K[.1]   ; FLOATING ONE
U 11D5, 0001,003C,0180,09D0,0000,11D6  ;2575  RC[0A]_D  ; EXPECTED AR LO
U 11D6, 0000,003C,1980,0800,0084,71D7  ;2576  SC_K[ZERO]  ; BIT POSITION COUNTER
U 11D7, 0018,0038,1D80,09B8,0000,1058  ;2577  RC[7]_SC  ; SAVE
U 1058, 0000,003D,0180,0800,0000,1153  ;2578 =0 SUBTEST
U 1059, 0000,003C,0180,0800,0000,105C  ;2579  NOP
U 105C, 0000,003D,0180,0800,0000,114E  ;2580 =0 ERLOOP
U 105D, 0000,003C,0180,0800,0000,1060  ;2581  NOP
          ;2582 =0
U 1060, 0000,003D,0180,0800,0000,119B  ;2583 LOOP7: CALL[TRP.8F]  ; TRAP TO ADDRESS 08F
U 1061, 0810,0038,0180,0950,0000,11D8  ;2584  D_RC[0A]  ; RESTORE D REGISTER
U 11D8, 0000,007C,0180,0800,0000,11D9  ;2585  CPSYNC   ; GO TO 0FA
U 11D9, 0000,007C,C580,3C00,0000,11DA  ;2586  ID[T1]_D,CPSYNC  ; LOAD AR, THEN GO TO 0FB
U 11DA, 0000,007C,01D8,0800,0000,11DB  ;2587  Q_ACC,CPSYNC  ; READ AR HI,THEN GO TO 0FC
U 11DB, 0011,2034,01C0,0940,0000,11DC  ;2588  Q_Q.AND.RC[8]  ; MASK OUT BITS 15,7
U 11DC, 0001,203C,0180,09D8,0000,11DD  ;2589  RC[0B]_Q  ; SAVE
U 11DD, 0000,003C,01D8,0800,0000,11DE  ;2590  Q_ACC   ; READ AR LO
U 11DE, 0001,203C,0180,09C8,0000,11DF  ;2591  RC[9]_Q  ; SAVE
U 11DF, 0011,2020,0180,0950,0010,11E0  ;2592  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK AR LO
U 11E0, 0000,013C,0180,0800,0000,1062  ;2593  Z?
U 1062, 0000,003D,0180,0800,0000,114F  ;2594 =0 ERROR1
U 1063, 0810,0038,0180,0950,0000,11E1  ;2595  D_RC[0A]  ; RESTORE  D REGISTER
U 11E1, 0811,0034,0180,0940,0000,11E2  ;2596  D_D.AND.RC[8]
U 11E2, 0001,003C,0180,09E0,0000,11E3  ;2597  RC[0C]_D  ; EXPECTED AR HI
U 11E3, 0010,0038,01C0,0958,0000,11E4  ;2598  Q_RC[0B]
U 11E4, 0011,2020,0180,0960,0010,11E5  ;2599  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK AR HI
U 11E5, 0000,013C,0180,0800,0000,1064  ;2600  Z?
U 1064, 0000,003D,0180,0800,0000,114F  ;2601 =0 ERROR1
U 1065, 0810,0038,0180,0950,0000,11E6  ;2602  D_RC[0A]  ; RESTORE D REGISTER
U 11E6, 0500,003C,0180,0800,0000,11E7  ;2603  D_D.LEFT  ; MOVE TO NEXT BIT POSITION
U 11E7, 0001,003C,0180,09D0,0000,11E8  ;2604  RC[0A]_D
U 11E8, 0000,003C,0180,0800,0080,D1E9  ;2605  SC_SC+1
U 11E9, 0018,0038,1D80,09B8,0000,11EA  ;2606  RC[7]_SC
U 11EA, 0000,003C,7580,0800,0014,B1EB  ;2607  EALU_SC-K[.20],CLK.UBCC  ; CHECK FOR 31 COUNTS
U 11EB, 0000,123C,0180,0800,0000,105B  ;2608  EALU.Z?
U 105B, 0000,003C,0180,0800,0000,1060  ;2609 =1011 J/LOOP7
U 105F, 0000,003C,0180,0800,0000,1066  ;2610 ENTST7: NOP
          ;2611 ;////////////////////////////////////////////
          ;2612 ;*
          ;2613 ; SUBTEST 2
          ;2614 ; TEST AR LATCH WITH FLOATING ZERO PATTERN
          ;2615 ;-
U 1066, 0000,003D,0180,0800,0000,1153  ;2616 =0 SUBTEST
U 1067, 0018,0038,0180,09F8,0000,1068  ;2617  RC[0F]_K[.8] ; 8 ARGUMENTS
U 1068, 2018,0039,0180,0800,4200,119E  ;2618 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 1069, 0818,0038,4580,0800,0000,11EC  ;2619  D_K[.8000]
U 11EC, 0819,0030,4180,0800,0000,11ED  ;2620  D_D.OR.K[.80]
U 11ED, 0001,0028,0180,09C0,0000,11EE  ;2621  RC[8]_NOT.D   ; SAVE MASK FOR AR HI
U 11EE, 0818,0038,0580,0800,0000,11EF  ;2622  D_K[.1]
U 11EF, 0001,003C,0180,09B0,0000,11F0  ;2623  RC[6]_D   ; SAVE
U 11F0, 0000,003C,1980,0800,0084,71F1  ;2624  SC_K[ZERO]  ; BIT POSITION COUNTER
```

```
U 11F1, 0018,0038,1D80,09B8,0000,106C  ;2625  RC[7]_SC  ; SAVE
U 106C, 0000,003D,0180,0800,0000,114E  ;2626 =0 ERLOOP
U 106D, 0F00,003C,0180,0800,0000,11F2  ;2627 LOOP7B: D_0
U 11F2, 0801,0028,0180,0800,0000,11F3  ;2628  D_NOT.D
U 11F3, 0811,0020,0180,0930,0000,11F4  ;2629  D_D.XOR.RC[6]  ; UPDATE FLOATING ZERO
U 11F4, 0001,003C,0180,09D0,0000,1070  ;2630  RC[0A]_D  ; EXPECTED AR LO
U 1070, 0000,003D,0180,0800,0000,119B  ;2631 =0 CALL[TRP.8F]  ; TRAP TO ADRESS 08F
U 1071, 0810,0038,0180,0950,0000,11F5  ;2632  D_RC[0A]  ; RESTORE D REGISTER
U 11F5, 0000,007C,0180,0800,0000,11F6  ;2633  CPSYNC  ; GO TO 0FA
U 11F6, 0000,007C,C580,3C00,0000,11F7  ;2634  ID[T1]_D,CPSYNC  ; LOAD AR, THEN GO TO 0FB
U 11F7, 0000,007C,01D8,0800,0000,11F8  ;2635  Q_ACC,CPSYNC  ; READ AR HI,THEN GO TO 0FC
U 11F8, 0011,2034,01C0,0940,0000,11F9  ;2636  Q_Q.AND.RC[8]  ; MASK OUT BITS 15,7
U 11F9, 0001,203C,0180,09D8,0000,11FA  ;2637  RC[0B]_Q  ; SAVE
U 11FA, 0000,003C,01D8,0800,0000,11FB  ;2638  Q_ACC  ; READ AR LO
U 11FB, 0001,203C,0180,09C8,0000,11FC  ;2639  RC[9]_Q  ; SAVE
U 11FC, 0011,2020,0180,0950,0010,11FD  ;2640  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK AR LO
U 11FD, 0000,013C,0180,0800,0000,1072  ;2641  Z?
U 1072, 0000,003D,0180,0800,0000,114F  ;2642 =0 ERROR1
U 1073, 0810,0038,0180,0950,0000,11FE  ;2643  D_RC[0A]  ; RESTORE  D REGISTER
U 11FE, 0811,0034,0180,0940,0000,11FF  ;2644  D_D.AND.RC[8]  ; MASK OUT BITS 15,7
U 11FF, 0001,003C,0180,09E0,0000,1300  ;2645  RC[0C]_D  ; EXPECTED AR HI
U 1300, 0010,0038,01C0,0958,0000,1301  ;2646  Q_RC[0B]
U 1301, 0011,2020,0180,0960,0010,1302  ;2647  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK AR HI
U 1302, 0000,013C,0180,0800,0000,1074  ;2648  Z?
U 1074, 0000,003D,0180,0800,0000,114F  ;2649 =0 ERROR1
U 1075, 0810,0038,0180,0930,0000,1303  ;2650  D_RC[6]  ; RESTORE D REGISTER
U 1303, 0500,003C,0180,0800,0000,1304  ;2651  D_D.LEFT  ; MOVE TO NEXT BIT POSITION
U 1304, 0001,003C,0180,09B0,0000,1305  ;2652  RC[6]_D  ; SAVE
U 1305, 0000,003C,0180,0800,0080,D306  ;2653  SC_SC+1
U 1306, 0018,0038,1D80,09B8,0000,1307  ;2654  RC[7]_SC
U 1307, 0000,003C,7580,0800,0014,B308  ;2655  EALU_SC-K[.20],CLK.UBCC  ; CHECK FOR 31 COUNTS
U 1308, 0000,123C,0180,0800,0000,106B  ;2656  EALU.Z?
U 106B, 0000,003C,0180,0800,0000,106D  ;2657 =1011 J/LOOP7B
U 106F, 0000,003C,0180,0800,0000,1076  ;2658 ENTST7B: NOP

          ;2659 ENTST7B: NOP

          ;2660 ;////////////////////////////////////////////
          ;2661 ;*
          ;2662 ; SUBTEST 3
          ;2663 ; TEST BR LATCH WITH FLOATING ONE PATTERN
          ;2664 ;-
U 1076, 0000,003D,0180,0800,0000,1153  ;2665 =0 SUBTEST
U 1077, 0018,0038,0180,09F8,0000,1078  ;2666  RC[0F]_K[.8]  ; 8 ARGUMENTS
U 1078, 2018,0039,0180,0800,4200,119E  ;2667 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 1079, 0818,0038,4D80,0800,0000,1309  ;2668  D_K[.FF00]
U 1309, 0819,0030,4180,0800,0000,130A  ;2669  D_D.OR.K[.80]  ; MASK OUT SIGN AND EXPONENT <15:07>
U 130A, 0001,0028,0180,09C0,0000,130B  ;2670  RC[8]_NOT.D  ; SAVE MASK
U 130B, 0818,0038,0580,0800,0000,130C  ;2671  D_K[.1]  ; FLOATING ONE
U 130C, 0001,003C,0180,09D0,0000,130D  ;2672  RC[0A]_D  ; EXPECTED BR, LO
U 130D, 0000,003C,1980,0800,0084,730E  ;2673  SC_K[ZERO]  ; BIT POSITION COUNTER
U 130E, 0018,0038,1D80,09B8,0000,107C  ;2674  RC[7]_SC  ; SAVE
U 107C, 0000,003D,0180,0800,0000,114E  ;2675 =0 ERLOOP
U 107D, 0000,003C,0180,0800,0000,1080  ;2676  NOP
          ;2677 =0
U 1080, 0000,003D,0180,0800,0000,119B  ;2678 LOOP7D: CALL[TRP.8F]  ; TRAP TO ADDRESS 08F
U 1081, 0810,0038,0180,0950,0000,130F  ;2679  D_RC[0A]  ; RESTORE D REGISTER
```

```
U 130F. 0000.007C.C580.3C00.0000.1082  ;2680  ID[T1]_D,CPSYNC  ; LOAD BR, GO TO 0FA
U 1082. 0000.003D.0D80.0800.1404.7196  ;2681 =0 STATE_K[.3],CALL[STEP] ; GO TO 0FD
U 1083. 0000.007C.01D8.0800.0000.1310  ;2682  Q_ACC,CPSYNC  ; READ BR HI,THEN GO TO 0FE
U 1310. 0011.2034.01C0.0940.0000.1311  ;2683  Q_Q.AND.RC[8]  ; MASK OUT <15:07>
U 1311. 0001.203C.0180.09D8.0000.1312  ;2684  RC[0B]_Q  ; RECEIVED BR HI
U 1312. 0000.003C.01D8.0800.0000.1313  ;2685  Q_ACC  ; READ BR LO
U 1313. 0001.203C.0180.09C8.0000.1314  ;2686  RC[9]_Q  ; SAVE
U 1314. 0011.2020.0180.0950.0010.1315  ;2687  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK BR LO
U 1315. 0000.013C.0180.0800.0000.1084  ;2688  Z?
U 1084. 0000.003D.0180.0800.0000.114F  ;2689 =0 ERROR1
U 1085. 0810.0038.0180.0950.0000.1316  ;2690  D_RC[0A]  ; RESTORE  D REGISTER
U 1316. 0811.0034.0180.0940.0000.1317  ;2691  D_D.AND.RC[8]
U 1317. 0001.003C.0180.09E0.0000.1318  ;2692  RC[0C]_D  ; EXPECTED BR HI
U 1318. 0810.0038.0180.0950.0000.1319  ;2693  D_RC[0A]  ; RESTORE D REGISTER
U 1319. 0010.0038.01C0.0958.0000.131A  ;2694  Q_RC[0B]
U 131A. 0011.2020.0180.0960.0010.131B  ;2695  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK BR HI
U 131B. 0000.013C.0180.0800.0000.1086  ;2696  Z?
U 1086. 0000.003D.0180.0800.0000.114F  ;2697 =0 ERROR1
U 1087. 0810.0038.0180.0950.0000.131C  ;2698  D_RC[0A]  ; RESTORE D REGISTER
U 131C. 0000.003C.01F8.0800.0000.131D  ;2699  Q_0
U 131D. 0500.003C.0180.0800.0000.131E  ;2700  D_D.LEFT  ; MOVE TO NEXT BIT POSITION
U 131E. 0001.003C.0180.09D0.0000.131F  ;2701  RC[0A]_D  ; BR LO EXPECTED
U 131F. 0000.003C.0180.0800.0080.D320  ;2702  SC_SC+1
U 1320. 0018.0038.1D80.09B8.0000.1321  ;2703  RC[7]_SC
U 1321. 0000.003C.7580.0800.0014.B322  ;2704  EALU_SC-K[.20],CLK.UBCC  ; CHECK FOR 31 COUNTS
U 1322. 0000.123C.0180.0800.0000.107B  ;2705  EALU.Z?
U 107B. 0000.003C.0180.0800.0000.1080  ;2706 =1011 J/LOOP7D
U 107F. 0000.003C.0180.0800.0000.1088  ;2707 ENTST7D: NOP
        ;2708 ;/////////////////////////////////////////
        ;2709 ;*
        ;2710 ; SUBTEST 4
        ;2711 ; TEST BR LATCH WITH FLOATING ZERO
        ;2712 ;-
U 1088. 0000.003D.0180.0800.0000.1153  ;2713 =0 SUBTEST
U 1089. 0018.0038.0180.09F8.0000.108C  ;2714  RC[0F]_K[.8]  ; 8 ARGUMENTS
U 108C. 2018.0039.0180.0800.4200.119E  ;2715 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 108D. 0818.0038.4D80.0800.0000.1323  ;2716  D_K[.FF00]
U 1323. 0819.0030.4180.0800.0000.1324  ;2717  D_D.OR.K[.80]  ; MASK OUT SIGN & EXPONENT <15:07>
U 1324. 0001.0028.0180.09C0.0000.1325  ;2718  RC[8]_NOT.D  ; SAVE MASK
U 1325. 0818.0038.0580.0800.0000.1326  ;2719  D_K[.1]
U 1326. 0001.003C.0180.09B0.0000.1327  ;2720  RC[6]_D  ; SAVE
U 1327. 0000.003C.1980.0800.0084.7328  ;2721  SC_K[ZERO]  ; BIT POSITION COUNTER
U 1328. 0018.0038.1D80.09B8.0000.1090  ;2722  RC[7]_SC  ; SAVE
U 1090. 0000.003D.0180.0800.0000.114E  ;2723 =0 ERLOOP
U 1091. 0F00.003C.0180.0800.0000.1329  ;2724 LOOP7E: D_0
U 1329. 0801.0028.0180.0800.0000.132A  ;2725  D_NOT.D
U 132A. 0811.0020.0180.0930.0000.132B  ;2726  D_D.XOR.RC[6]  ; UP DATE FLOATING ZERO
U 132B. 0001.003C.0180.09D0.0000.1092  ;2727  RC[0A]_D  ; EXPECTED BR LO
U 1092. 0000.003D.0180.0800.0000.119B  ;2728 =0 CALL[TRP.8F]  ; TRAP TO ADDRESS 08F
U 1093. 0810.0038.0180.0950.0000.132C  ;2729  D_RC[0A]  ; RESTORE D REGISTER
U 132C. 0000.007C.C580.3C00.0000.1094  ;2730  ID[T1]_D,CPSYNC  ; LOAD BR, GO TO 0FA
U 1094. 0000.003D.0D80.0800.1404.7196  ;2731 =0 STATE_K[.3],CALL[STEP] ; GO TO 0FD
U 1095. 0000.007C.01D8.0800.0000.132D  ;2732  Q_ACC,CPSYNC  ; READ BR HI,THEN GO TO 0FE
U 132D. 0011.2034.01C0.0940.0000.132E  ;2733  Q_Q.AND.RC[8]  ; MASK OUT <15:07>
U 132E. 0001.203C.0180.09D8.0000.132F  ;2734  RC[0B]_Q  ; RECEIVED BR HI
```

```
U 132F, 0000.003C.01D8.0800.0000.1330  ;2735  Q_ACC   ; READ BR LO
U 1330, 0001.203C.0180.09C8.0000.1331  ;2736  RC[9]_Q   ; SAVE
U 1331, 0011.2020.0180.0950.0010.1332  ;2737  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK BR LO
U 1332, 0000.013C.0180.0800.0000.1096  ;2738  Z?
U 1096, 0000.003D.0180.0800.0000.114F  ;2739 =0 ERROR1
U 1097, 0810.0038.0180.0950.0000.1333  ;2740  D_RC[0A]  ; RESTORE D REGISTER
U 1333, 0811.0034.0180.0940.0000.1334  ;2741  D_D.AND.RC[8]
U 1334, 0001.003C.0180.09E0.0000.1335  ;2742  RC[0C]_D  ; EXPECTED BR HI
U 1335, 0810.0038.0180.0950.0000.1336  ;2743  D_RC[0A]  ; RESTORE D REGISTER
U 1336, 0010.0038.01C0.0958.0000.1337  ;2744  Q_RC[0B]
U 1337, 0011.2020.0180.0960.0010.1338  ;2745  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK BR HI
U 1338, 0000.013C.0180.0800.0000.1098  ;2746  Z?
U 1098, 0000.003D.0180.0800.0000.114F  ;2747 =0 ERROR1
U 1099, 0810.0038.0180.0930.0000.1339  ;2748  D_RC[6]   ; RESTORE D REGISTER
U 1339, 0500.003C.0180.0800.0000.133A  ;2749  D_D.LEFT   ; MOVE TO NEXT BIT POSITION
U 133A, 0001.003C.0180.09B0.0000.133B  ;2750  RC[6]_D   ; SAVE
U 133B, 0000.003C.0180.0800.0080.D33C  ;2751  SC_SC+1
U 133C, 0018.0038.1D80.09B8.0000.133D  ;2752  RC[7]_SC
U 133D, 0000.003C.7580.0800.0014.B33E  ;2753  EALU_SC-K[.20],CLK.UBCC  ; CHECK FOR 31 COUNTS
U 133E, 0000.123C.0180.0800.0000.108B  ;2754  EALU.Z?
U 108B, 0000.003C.0180.0800.0000.1091  ;2755 =1011 J/LOOP7E
U 108F, 0000.003C.0180.0800.0000.109C  ;2756 ENTST7E: NOP
           ;2757 ;X8
           ;2758 ;$ 5060,0040
           ;2759 ;
```

```
                ;2760  .PAGE "TEST 22F FPA SCRATCH PAD COPIES DATA TEST 1"
                ;2761 ;*****************************************************************
                ;2762 ;**
                ;2763 ;  FPA SCRATCH PAD COPIES DATA TEST 1
                ;2764 ;
                ;2765 ; TEST DESCRIPTION AN ALL ZEROS/ONES TEST PATTERN IS USED AS
                ;2766 ;   DESCRIBED IN THE FOLLOWING TEST SEQUENCE.
                ;2767 ;
                ;2768 ; TEST SEQUENCE  DESCRIPTION
                ;2769 ; (1)  GENERATE TEST PATTERN
                ;2770 ; (2)  WRITE TEST PATTERN INTO THE SCRATCH PAD
                ;2771 ; (3)  READ BACK AND VERIFY RESULT
                ;2772 ; (4)  REPEAT STEPS 2,3,4 UNTILL ALL SCRATCH PADS
                ;2773 ;   HAVE BEEN VERIFIED.
                ;2774 ;
                ;2775 ; FPA UCODE USED
                ;2776 ; LOCATION CONTENTS
                ;2777 ; 081: BSC/R,SCR/CPU,FADC/LD,WAIT,FPSYNC,NAD/082
                ;2778 ; 082: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/082
                ;2779 ;
                ;2780 ; LOGIC DESCRIPTION
                ;2781 ; THE RA, RB SCRATCH PADS (GENERAL REGISTERS) ARE ON
                ;2782 ; THE FNM MODULE. THE SCRATCH PAD ADDRESSING LOGIC IS
                ;2783 ; ON THE FCT MODULE.
                ;2784 ;
                ;2785 ; ERROR DESCRIPTION
                ;2786 ; EXPECTED TRAP UADDRESS
                ;2787 ; RECEIVED TRAP UADDRESS
                ;2788 ; EXPECTED RA, RB SCRATCH PAD DATA
                ;2789 ; RECEIVED RA SCRATCH PAD DATA
                ;2790 ; RECEIVED RB SCRATCH PAD DATA
                ;2791 ; SCRATCH PAD COUNTER
                ;2792 ; LOOP FLAG
                ;2793 ; ERROR1 COUNTER
                ;2794 ;
                ;2795 ;
                ;2796 ; SYNC POINT DESCRIPTION
                ;2797 ;--
                ;2798 ;*****************************************************************
                ;2799 ;/////////////////////////////////////////
                ;2800 ;*
                ;2801 ; SUBTEST 1
                ;2802 ; TEST FPA SCRATCH PAD COPIES WITH ALL ZEROS/ONES PATTERN
                ;2803 ;-
                ;2804 =0
U 109C, 0018,0039,5D80,09F8,0000,108A  ;2805 BTST9: NEWTST[.7]
U 109D, 0810,0038,0180,0978,0000,133F  ;2806  D_RC[0F]
U 133F, 0019,0014,0580,09F8,0000,1340  ;2807  RC[0F]_D+K[.1]
U 1340, 0018,0038,0980,09C0,0000,10A0  ;2808  RC[8]_K[.2]  ; LOOP FLAG
U 10A0, 2018,0039,0180,0800,4200,119E  ;2809 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD]  ; ADDD R,R
U 10A1, 0000,003C,1980,0800,0084,7341  ;2810  SC_K[ZERO]  ; START WITH R0
U 1341, 0018,0038,1D80,09C8,0000,1342  ;2811  RC[9]_SC  ; SAVE
U 1342, 0F00,003C,0180,0800,0000,1343  ;2812  D_0  ; PATTERN IS ZERO
U 1343, 0001,003C,0180,09E0,0000,10A2  ;2813  RC[0C]_D  ; EXPECTED RA, RB
U 10A2, 0000,003D,0180,0800,0000,114E  ;2814 =0 ERLOOP
```

```
U 10A3. 0000.003C.0180.0800.0000.10A4  ;2815  NOP
        ;2816 =0
U 10A4. 0000.003D.0180.0800.0000.1199  ;2817 LOOP8: CALL[TRP.81]  ; TRAP TO 081
U 10A5. 0810.0038.0180.0960.0000.1344  ;2818  D_RC[0C]  ; RESTORE D REGISTER
U 1344. 0001.003C.0180.08E8.0000.1345  ;2819  R(SC)_D   ; LOAD SCRATCH PAD
U 1345. 0000.003C.0180.0868.0000.1346  ;2820  LAB_R(SC)  ; LOAD FPA GR LA,LB
U 1346. 0000.007C.01D8.0868.0000.1347  ;2821  Q_ACC,CPSYNC,LAB_R(SC) ; READ RA THEN GO TO 082
U 1347. 0001.203C.0180.09D8.0000.1348  ;2822  RC[0B]_Q  ; RECEIVED RA
U 1348. 0000.003C.01D8.0800.0000.1349  ;2823  Q_ACC   ; READ RB
U 1349. 0001.203C.0180.09D0.0000.134A  ;2824  RC[0A]_Q  ; RECEIVED RB
U 134A. 0011.2020.0180.0960.0010.134B  ;2825  ALU_Q.XOR.RC[0C],CLK.UBCC
U 134B. 0000.013C.0180.0800.0000.10A6  ;2826  Z?
U 10A6. 0018.0039.0580.09B8.0000.114F  ;2827 =0 ERROR1,RC[7]_K[.1]
U 10A7. 0810.0038.0180.0960.0000.134C  ;2828  D_RC[0C]  ; RESTORE D REGISTER
U 134C. 0010.0038.01C0.0958.0000.134D  ;2829  Q_RC[0B]  ; RECEIVED RA
U 134D. 0011.2020.0180.0960.0010.134E  ;2830  ALU_Q.XOR.RC[0C],CLK.UBCC
U 134E. 0000.013C.0180.0800.0000.10A8  ;2831  Z?
U 10A8. 0018.0039.0980.09B8.0000.114F  ;2832 =0 ERROR1,RC[7]_K[.2]
U 10A9. 0810.0038.0180.0960.0000.134F  ;2833  D_RC[0C]  ; RESTORE D REGISTER
U 134F. 0000.003C.0180.0800.0080.D350  ;2834  SC_SC+1   ; UPDATE REGISTER COUNTER
U 1350. 0018.0038.1D80.09C8.0000.1351  ;2835  RC[9]_SC  ; SAVE COUNTER
U 1351. 0000.003C.6580.0800.0014.B352  ;2836  EALU_SC-K[.10],CLK.UBCC  ; ALL SCRATCH PADS CHECKED?
U 1352. 0000.123C.0180.0800.0000.109B  ;2837  EALU.Z?
U 109B. 0000.003C.0180.0800.0000.10A4  ;2838 =1011 J/LOOP8
U 109F. 0000.003C.1980.0800.0084.7353  ;2839  SC_K[ZERO] ; RE-INITIALIZE SC
U 1353. 0801.0028.0180.0800.0000.1354  ;2840  D_NOT.D  ; PATTERN IS ONES
U 1354. 0001.003C.0180.09E0.0000.1355  ;2841  RC[0C]_D ; SAVE
U 1355. 0810.0038.0180.0940.0000.1356  ;2842  D_RC[8]  ; LOOP FLAG
U 1356. 0019.0000.0580.09C0.0010.1357  ;2843  RC[8]_D-K[.1],CLK.UBCC
U 1357. 0000.013C.0180.0800.0000.10AC  ;2844  Z?
U 10AC. 0000.003C.0180.0800.0000.10A4  ;2845 =0 J/LOOP8
U 10AD. 0000.003C.0180.0800.0000.10B0  ;2846 ENTST8: NOP
```

```
                ;2847  .PAGE "TEST 230 FPA SCRATCH PADS DATA TEST 2"
                ;2848 ;*****************************************************************
                ;2849 ;**
                ;2850 ; FPA SCRATCH PADS DATA INTEGRITY TEST 2
                ;2851 ;
                ;2852 ; TEST DESCRIPTION A FLOATING ZEROS/ONES TEST PATTERN IS USED AS
                ;2853 ;   DESCRIBED IN THE FOLOWING TEST SEQUENCE.
                ;2854 ;
                ;2855 ; TEST SEQUENCE  DESCRIPTION
                ;2856 ; (1)  GENERATE TEST PATTERN
                ;2857 ; (2)  WRITE TEST PATTERN INTO THE SCRATCH PAD
                ;2858 ; (3)  READ BACK AND VERIFY RESULT
                ;2859 ; (4)  REPEAT STEPS 2,3,4 UNTILL ALL BITS OF
                ;2860 ;   EACH SCRATCH PAD HAS BEEN VERIFIED.
                ;2861 ;
                ;2862 ; FPA UCODE USED
                ;2863 ; LOCATION CONTENTS
                ;2864 ; 081: BSC/R,SCR/CPU,FADC/LD,WAIT,FPSYNC,NAD/082
                ;2865 ; 082: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/082
                ;2866 ;
                ;2867 ; LOGIC DESCRIPTION
                ;2868 ; THE RA, RB SCRATCH PADS (GENERAL REGISTERS) ARE ON
                ;2869 ; THE FNM MODULE. THE SCRATCH PAD ADDRESSING LOGIC IS
                ;2870 ; ON THE FCT MODULE.
                ;2871 ;
                ;2872 ; ERROR DESCRIPTION
                ;2873 ; EXPECTED TRAP UADDRESS
                ;2874 ; RECEIVED TRAP UADDRESS
                ;2875 ; EXPECTED RA, RB SCRATCH PAD DATA
                ;2876 ; RECEIVED RA SCRATCH PAD DATA
                ;2877 ; RECEIVED RB SCRATCH PAD DATA
                ;2878 ; SCRATCH PAD COUNTER
                ;2879 ; LOOP FLAG
                ;2880 ; BIT COUNTER
                ;2881 ;
                ;2882 ; SYNC POINT DESCRIPTION
                ;2883 ;--
                ;2884 ;*****************************************************************
                ;2885 ;/////////////////////////////////////////
                ;2886 ;*
                ;2887 ; TEST FPA SCRATCH PAD COPIES WITH FLOATING ONE/ZERO PATTERN
                ;2888 ;-
                ;2889 =0
U 10B0, 0018.0039.0180.09F8.0000.108A  ;2890 BTST10: NEWTST[.8]
U 10B1, 0018.0038.0980.09C0.0000.10B2  ;2891  RC[8]_K[.2]  ; LOOP FLAG
U 10B2, 2018.0039.0180.0800.4200.119E  ;2892 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; ADDD R,R
U 10B3, 0000.003C.1980.0800.0084.7358  ;2893  SC_K[ZERO]  ; START WITH R0
U 1358, 0018.0038.1D80.09B8.0000.1359  ;2894  RC[7]_SC  ; SAVE SC FOR BIT COUNTER
U 1359, 0803.001C.0180.0800.0000.135A  ;2895  D_NOT.MASK  ; FLOATING ONE
U 135A, 0001.003C.0180.09E0.0000.10B4  ;2896  RC[0C]_D
U 10B4, 0000.003D.0180.0800.0000.114E  ;2897 =0 ERLOOP
U 10B5, 0000.003C.0180.0800.0000.10B6  ;2898  NOP
                ;2899 =0
U 10B6, 0000.003D.0180.0800.0000.1199  ;2900 LOOP8A: CALL[TRP,81]  ; TRAP TO 081
U 10B7, 0810.0038.0180.0960.0000.135B  ;2901  D_RC[0C]  ; RESTORE D REGISTER
```

```
U 135B, 0001,003C,0180,08E8,0000,135C  ;2902  R(SC)_D   ; LOAD CPU/FPA SCRATCH PAD
U 135C, 0000,003C,0180,0868,0000,135D  ;2903  LAB_R(SC)  ; LOAD FPA GR LA,LB
U 135D, 0000,007C,01D8,0868,0000,135E  ;2904  Q_ACC,CPSYNC,LAB_R(SC) ; READ RA THEN GO TO 082
U 135E, 0001,203C,0180,09D8,0000,135F  ;2905  RC[0B]_Q  ; RECEIVED RA
U 135F, 0000,003C,01D8,0800,0000,1360  ;2906  Q_ACC   ; READ RB
U 1360, 0001,203C,0180,09D0,0000,1361  ;2907  RC[0A]_Q  ; RECEIVED RB
U 1361, 0011,2020,0180,0960,0010,1362  ;2908  ALU_Q.XOR.RC[0C],CLK.UBCC
U 1362, 0000,013C,0180,0800,0000,10B8  ;2909  Z?
U 10B8, 0000,003D,0180,0800,0000,114F  ;2910 =0 ERROR1
U 10B9, 0810,0038,0180,0960,0000,1363  ;2911  D_RC[0C]  ; RESTORE D REGISTER
U 1363, 0010,0038,01C0,0958,0000,1364  ;2912  Q_RC[0B]  ; RECEIVED RA
U 1364, 0011,2020,0180,0960,0010,1365  ;2913  ALU_Q.XOR.RC[0C],CLK.UBCC
U 1365, 0000,013C,0180,0800,0000,10BC  ;2914  Z?
U 10BC, 0000,003D,0180,0800,0000,114F  ;2915 =0 ERROR1
U 10BD, 0810,0038,0180,0960,0000,1366  ;2916  D_RC[0C]  ; RESTORE D REGISTER
U 1366, 0000,003C,0180,0800,0080,D367  ;2917  SC_SC+1   ; UPDATE REGISTER COUNTER
U 1367, 0018,0038,1D80,09C8,0000,1368  ;2918  RC[9]_SC  ; SAVE COUNTER
U 1368, 0000,003C,6580,0800,0014,B369  ;2919  EALU_SC-K[.10],CLK.UBCC  ; ALL SCRATCH PADS CHECKED?
U 1369, 0000,123C,0180,0800,0000,10AB  ;2920  EALU.Z?
U 10AB, 0000,003C,0180,0800,0000,10B6  ;2921 =1011 J/LOOP8A
U 10AF, 0010,0038,0180,0938,0082,136A  ;2922  SC_RC[7]  ; RESTORE SC FOR BIT COUNTER
U 136A, 0000,003C,0180,0800,0080,D36B  ;2923  SC_SC+1   ; UP DATE BIT COUNTER
U 136B, 0018,0038,1D80,09B8,0000,136C  ;2924  RC[7]_SC  ; SAVE BIT COUNTER
U 136C, 0810,0038,0180,0940,0000,136D  ;2925  D_RC[8]   ; LOOP FLAG
U 136D, 0019,0000,0580,0800,0010,136E  ;2926  ALU_D-K[.1],CLK.UBCC ; CHECK LOOP FLAG
U 136E, 0000,013C,0180,0800,0000,10C0  ;2927  Z?
U 10C0, 0000,003C,0180,0800,0000,136F  ;2928 =0 J/FLOAT1
U 10C1, 0810,0038,0580,0960,1404,70C2  ;2929 FLOAT0: D_RC[0C],STATE_K[.1] ; SET STATE COUNTER FOR PATTERN UPDATE
U 10C2, 0000,003D,0180,0800,0000,1181  ;2930 =0 CALL[SHIFQD]  ; PATTERN UPDATE
U 10C3, 0000,003C,0180,0800,0000,1371  ;2931  J/CHKBIT
U 136F, 0810,0038,0180,0960,0000,1370  ;2932 FLOAT1: D_RC[0C]
U 1370, 0500,003C,0180,0800,0000,1371  ;2933  D_D.LEFT
U 1371, 0001,003C,0180,09E0,0000,1372  ;2934 CHKBIT: RC[0C]_D  ; SAVE UPDATED PATTERN
U 1372, 0010,0038,0180,0938,0082,1373  ;2935  SC_RC[7]  ; RESTORE SC TO BIT COUNTER
U 1373, 0000,003C,7580,0800,0014,B374  ;2936  EALU_SC-K[.20],CLK.UBCC ; CHECK BIT COUNTER
U 1374, 0000,123C,0180,0800,0000,10BB  ;2937  EALU.Z?
U 10BB, 0000,003C,1980,0800,0084,70B6  ;2938 =1011 SC_K[ZERO],J/LOOP8A ; RESTORE TO REGISTER COUNTER AND RETURN
U 10BF, 0003,003C,0180,09B8,0000,1375  ;2939  RC[7]_0   ; RE-INITIALIZE RC[7] TO BIT COUNTER
U 1375, 0810,0038,0180,0940,0000,1376  ;2940  D_RC[8]   ; LOOP FLAG
U 1376, 0019,0000,0580,09C0,0010,1377  ;2941  RC[8]_D-K[.1],CLK.UBCC ; DECREMENT & CHECK LOOP FLAG
U 1377, 0000,013C,1980,0800,0084,70C4  ;2942  Z?,SC_K[ZERO]
U 10C4, 0000,0038,0180,09E0,0000,10B6  ;2943 =0 RC[0C]_MASK,J/LOOP8A ; RESTORE SC TO REGISTER COUNTER
U 10C5, 0000,003C,0180,0800,0000,10C6  ;2944 ENTST8A: NOP
```

```
                ;2945  .PAGE "TEST 231 ADDRESSING TEST FOR THE FPA SCRATCH PAD COPIES"
                ;2946 ;*****************************************************************
                ;2947 ;**
                ;2948 ; ADDRESSING TEST FOR THE FPA SCRATCH PAD COPIES
                ;2949 ;
                ;2950 ; TEST DESCRIPTION
                ;2951 ; EACH NIBBLE IN EACH SCRATCH PAD IS WRITTEN WITH THE
                ;2952 ; ADDRESS OF THE SCRATCH PAD. EACH SCRATCH PAD IS THEN
                ;2953 ; READ BACK AND VERIFIED.
                ;2954 ;
                ;2955 ; FPA UCODE USED
                ;2956 ; LOCATION CONTENTS
                ;2957 ; 081: BSC/R,SCR/CPU,FADC/LD,WAIT,FPSYNC,NAD/082
                ;2958 ; 082: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/082
                ;2959 ;
                ;2960 ; LOGIC DESCRIPTION
                ;2961 ; THE RA, RB SCRATCH PADS (GENERAL REGISTERS) ARE ON
                ;2962 ; THE FNM MODULE. THE SCRATCH PAD ADDRESSING LOGIC IS
                ;2963 ; ON THE FCT MODULE.
                ;2964 ;
                ;2965 ; ERROR DESCRIPTION
                ;2966 ; EXPECTED TRAP UADDRESS
                ;2967 ; RECEIVED TRSP UADDRESS
                ;2968 ; EXPECTED SCRATCH PAD DATA
                ;2969 ; RECEIVED RA SCRATCH PAD DATA
                ;2970 ; RECEIVED RB SCRATCH PAD DATA
                ;2971 ; SCRATCH PAD ADDRESS COUNTER
                ;2972 ;
                ;2973 ; SYNC POINT DESCRIPTION
                ;2974 ;--
                ;2975 ;*****************************************************************
                ;2976 ;/////////////////////////////////////////////
                ;2977 ;*
                ;2978 ; ADDRESSING TEST OF THE FPA SCRATCH  PAD COPIES
                ;2979 ;-
                ;2980 =0
U 10C6. 0000.003D.0180.0800.0000.108A  ;2981 BTST11: NEWTST
U 10C7. 0818.0038.0D80.0800.0000.1378  ;2982  D_K[.3]
U 1378. 0021.003C.0180.09F8.0000.10C8  ;2983  RC[0F]_D.LEFT  ; 6 ARGUMENTS
U 10C8. 2018.0039.0180.0800.4200.119E  ;2984 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; LOAD CODE
U 10C9. 0000.003C.1980.0800.0084.7379  ;2985  SC_K[ZERO]  ; START WITH R0
U 1379. 0018.0038.1D80.09C8.0000.137A  ;2986  RC[9]_SC  ; SAVE SC FOR REGISTER COUNTER
U 137A. 0F00.003C.0180.0800.0000.137B  ;2987  D_0   ; FIRST ADDRESS
U 137B. 0001.003C.0180.09E0.0000.10CC  ;2988  RC[0C]_D  ; SAVE
U 10CC. 0000.003D.0180.0800.0000.114E  ;2989 =0 ERLOOP
U 10CD. 0000.003C.0180.0800.0000.10D0  ;2990  NOP
                ;2991 =0
U 10D0. 0000.003D.0180.0800.0000.1199  ;2992 LOOP8B: CALL[TRP.81]  ; TRAP TO 81
U 10D1. 0810.0038.0180.0960.0000.137C  ;2993  D_RC[0C]  ; RESTORE D REGISTER
U 137C. 0001.003C.0180.08E8.0000.137D  ;2994  R(SC)_D   ; LOAD SCRATCH PAD
U 137D. 0000.003C.0180.0800.0080.D37E  ;2995  SC_SC+1  ; UPDATE ADDRESS COUNTER
U 137E. 0018.0038.1D80.09C8.0000.10D2  ;2996  RC[9]_SC  ; SAVE
U 10D2. 0819.0015.0580.0800.0000.1186  ;2997 =0 D_D+K[.1],CALL[SHIFD4] ; UPDATE ADDRESS
U 10D3. 0001.003C.0180.09E0.0000.137F  ;2998  RC[0C]_D  ; SAVE
U 137F. 0000.003C.6580.0800.0014.B380  ;2999  EALU_SC-K[.10],CLK.UBCC
```

```
U 1380. 0000.123C.0180.0800.0000.10CB  ;3000  EALU.Z?
U 10CB. 0000.003C.0180.0800.0000.10D0  ;3001 =1011 J/LOOP8B
U 10CF. 0000.003C.1980.0800.0084.7381  ;3002  SC_K[ZERO]  ; RE-INITIALIZE REGISTER COUNTER
U 1381. 0018.0038.1D80.09C8.0000.10D4  ;3003  RC[9]_SC  ; SAVE
        ;3004 =0
U 10D4. 0000.003D.0180.0800.0000.1199  ;3005 VRFY: CALL[TRP.81]  ; TRAP TO 081
U 10D5. 0000.003C.0180.0868.0000.1382  ;3006  LAB_R(SC)
U 1382. 0000.007C.01D8.0868.0000.1383  ;3007  Q_ACC,CPSYNC,LAB_R(SC) ; READ RA THEN GO TO 082
U 1383. 0001.203C.0180.09D8.0000.1384  ;3008  RC[0B]_Q  ; RECEIVED RA
U 1384. 0000.003C.01D8.0800.0000.1385  ;3009  Q_ACC  ; READ RB
U 1385. 0001.203C.0180.09D0.0000.1386  ;3010  RC[0A]_Q  ; RECEIVED RB
U 1386. 000D.2020.0180.0868.0010.1387  ;3011  ALU_Q.XOR.R(SC),CLK.UBCC
U 1387. 0010.0138.0180.09E0.0000.10D6  ;3012  Z?,RC[0C]_LC
U 10D6. 0000.003D.0180.0800.0000.114F  ;3013 =0 ERROR1
U 10D7. 0010.0038.01C0.0958.0000.1388  ;3014  Q_RC[0B]  ; RECEIVED RA
U 1388. 000D.2020.0180.0868.0010.1389  ;3015  ALU_Q.XOR.R(SC),CLK.UBCC
U 1389. 0000.013C.0180.0800.0000.10D8  ;3016  Z?
U 10D8. 0000.003D.0180.0800.0000.114F  ;3017 =0 ERROR1
U 10D9. 0000.003C.0180.09E0.0000.1004  ;3018  RC[0C]_LA  ; EXPECTED RA,RB
U 1004. 0000.003D.0180.0800.0000.118D  ;3019 =00 CALL[SC.CHK]  ; CHECK REGISTER COUNTER
U 1005. 0018.0038.1D80.09C8.0000.10D4  ;3020  RC[9]_SC,J/VRFY
U 1006. 0000.003C.0180.0800.0000.1007  ;3021  NOP
U 1007. 0000.003C.0180.0800.0000.10DA  ;3022  NOP
```

```
                ;3023  .PAGE "TEST 232 SP1 ON RA, GENERAL REGISTERS ADDRESSING TEST 1"
                ;3024 ;*****************************************************************
                ;3025 ;++
                ;3026 ; SP1 ON RA, GENERAL REGISTERS ADDRESSING TEST 1
                ;3027 ;
                ;3028 ; TEST DESCRIPTION
                ;3029 ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;3030 ;  SET COPY IN THE FPA. TESTED HERE IS THE SPECIFIER 1
                ;3031 ;  ADDRESS FOR THE RA REGISTERS WHICH COMES FROM THE
                ;3032 ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;3033 ;  WITH A UNIQUE DATA PATTERN:
                ;3034 ;   R[0] 00000000
                ;3035 ;   R[1] 11111111
                ;3036 ;   R[2] 22222222
                ;3037 ;    ...
                ;3038 ;   R[0F] FFFFFFFF
                ;3039 ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;3040 ;  USING THE IB SP1.
                ;3041 ;
                ;3042 ; FPA UCODE USED
                ;3043 ;  LOCATION CONTENTS
                ;3044 ;  13:  BSC/R,SCR/R.R,FADC/LD,NAD/82
                ;3045 ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;3046 ;
                ;3047 ; LOGIC DESCRIPTION
                ;3048 ;
                ;3049 ; ERROR DESCRIPTION
                ;3050 ;  EXPECTED DATA READ FROM REGISTER SP1
                ;3051 ;  GOT DATA READ FROM REGISTER SP1
                ;3052 ;  SP1 REGISTER ADDRESS
                ;3053 ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP1
                ;3054 ;
                ;3055 ; SYNC POINT DESCRIPTION
                ;3056 ;
                ;3057 ;--
                ;3058 ;*****************************************************************
                ;3059 =0
U 10DA, 0000.003D.0180.0800.0000.108A  ;3060 RABTA: NEWTST
U 10DB, 0018.0038.1180.09F8.0000.10DC  ;3061  RC[0F]_K[.4]
U 10DC, 0018.0039.6180.09E0.0000.116E  ;3062 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;3063      ; COUNTER AND PUT THE DATA INTO
                ;3064      ; THE REGISTERS.
U 10DD, 0000.003C.1180.0800.0084.738A  ;3065  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 138A, 0818.0038.79F8.0800.0000.138B  ;3066  D_K[.30],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 138B, 0D00.003C.0180.0800.0000.138C  ;3067  D_DAL.SC
U 138C, 0001.003C.3180.0980.0000.138D  ;3068  RC[0]_D,KMX/.40  ; RC[0]=300
U 138D, 0819.0014.3180.09D8.0000.138E  ;3069  ALU_D+K[.40],D_ALU,RC[0B]_ALU ; RC[0B]=340
U 138E, 0019.0014.3180.0988.0000.138F  ;3070  ALU_D+K[.40],RC[1]_ALU ; RC[1]=380
U 138F, 0818.0038.6580.0800.0000.10DE  ;3071  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 10DE, 0819.0015.0D80.0800.0000.115E  ;3072 =0 ALU_D+K[.3],D_ALU,CALL,J/GENTRA
U 10DF, 0001.003C.0180.0990.0000.10E0  ;3073  RC[2]_D
U 10E0, 0000.003D.0180.0800.0000.114E  ;3074 =0 ERLOOP
U 10E1, 0000.003C.0180.0800.0000.10E2  ;3075  NOP
                ;3076 =0
U 10E2, 0000.003D.0180.0800.0000.1159  ;3077 RABTA1: CALL,J/FPINIT
```

```
U 10E3, 0010.0038.0180.0908.0200.10E4  ;3078  VA_RC[1]  ; LOAD THE PRN REGISTER WITH
U 10E4, 0000.003D.0180.0800.0000.1164  ;3079 =0 CALL,J/LOADIB  ; ADDRESS OF
U 10E5, 0000.003C.0180.0800.0000.1010  ;3080  NOP
U 1010, 0000.003D.0180.0800.0000.1011  ;3081 =00 CALL
U 1011, 0000.003F.0180.0800.0000.1200  ;3082  SUB/SPEC,J/IRTAB
U 1012, 0000.003C.0180.0800.0000.1013  ;3083  NOP
U 1013, 0010.0038.0180.0958.0200.10E6  ;3084  VA_RC[0B]  ; LOAD THE OPCODE WITH THE
U 10E6, 0000.003D.0180.0800.0000.1164  ;3085 =0 CALL,J/LOADIB  ; SP1 BEING TESTED INTO THE IB.
U 10E7, 0810.0038.0180.0910.0000.1390  ;3086  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 1390, 0000.003C.5980.3C00.0000.1391  ;3087  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 1391, 0000.00FC.0380.0800.0000.1392  ;3088  TRAP.FPA
U 1392, 0000.003C.01D8.0A78.0000.1393  ;3089  Q_ACC,LAB_R[0F]  ; GET THE RESULT.
U 1393, 0010.0038.0180.0960.0082.1394  ;3090  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 1394, 0800.003C.0180.0868.0000.1395  ;3091  ALU_R(SC),D_ALU
U 1395, 0001.003C.0180.09F0.0000.1396  ;3092  RC[0E]_D
U 1396, 0001.203C.0180.09E8.0000.1397  ;3093  RC[0D]_Q
U 1397, 001D.0000.0180.0800.0010.1398  ;3094  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 1398, 0000.013C.0180.0800.0000.10E8  ;3095  Z?
U 10E8, 0000.003D.0180.0800.0000.114F  ;3096 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 10E9, 0810.0038.0180.0958.0000.1399  ;3097  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP1.
U 1399, 0019.0014.1180.09D8.0000.139A  ;3098  ALU_D+K[.4],RC[0B]_ALU
U 139A, 0810.0038.0180.0960.0010.139B  ;3099  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 139B, 0000.013C.0180.0800.0000.10EA  ;3100  Z?
U 10EA, 0019.0000.0580.09E0.0000.10E2  ;3101 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTA1 ; NO SO CONTINUE.
U 10EB, 0000.003C.0180.0800.0000.10EC  ;3102  NOP  ; DONE
          ;3103 ;
          ;3104 ; THIS TABLE IS USED BY ALL THE REGISTER ADDRESSING TESTS
          ;3105 ; THAT DEPEND UPON THE SPECIFIERS FROM THE IB.
          ;3106 ;
          ;3107 ;x 300
          ;3108 ;
          ;3109 ; SP2 DATA
          ;3110 ;
          ;3111 ;$ 5060,005F
          ;3112 ;$ 5060,005E
          ;3113 ;$ 5060,005D
          ;3114 ;$ 5060,005C
          ;3115 ;$ 5060,005B
          ;3116 ;$ 5060,005A
          ;3117 ;$ 5060,0059
          ;3118 ;$ 5060,0058
          ;3119 ;$ 5060,0057
          ;3120 ;$ 5060,0056
          ;3121 ;$ 5060,0055
          ;3122 ;$ 5060,0054
          ;3123 ;$ 5060,0053
          ;3124 ;$ 5060,0052
          ;3125 ;$ 5060,0051
          ;3126 ;$ 5060,0050
          ;3127 ;
          ;3128 ;x340
          ;3129 ;
          ;3130 ; SP1 DATA
          ;3131 ;
          ;3132 ;$ 5F60,0050
```

```
;3133 ;$ 5E60,0050
;3134 ;$ 5D60,0050
;3135 ;$ 5C60,0050
;3136 ;$ 5B60,0050
;3137 ;$ 5A60,0050
;3138 ;$ 5960,0050
;3139 ;$ 5860,0050
;3140 ;$ 5760,0050
;3141 ;$ 5660,0050
;3142 ;$ 5560,0050
;3143 ;$ 5460,0050
;3144 ;$ 5360,0050
;3145 ;$ 5260,0050
;3146 ;$ 5160,0050
;3147 ;$ 5060,0050
;3148 ;
;3149 ;x380
;3150 ;
;3151 ;$ 5F60,005F
;3152 ;
;3153
```

```
                ;3154  .PAGE "TEST 233 SP2 ON RB, GENERAL REGISTERS ADDRESSING TEST 2"
                ;3155 ;*****************************************************************
                ;3156 ;**
                ;3157 ; SP2 ON RB, GENERAL REGISTERS ADDRESSING TEST 2
                ;3158 ;
                ;3159 ; TEST DESCRIPTION
                ;3160 ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;3161 ;  SET COPY IN THE FPA. TESTED HERE IS THE SPECIFIER 2
                ;3162 ;  ADDRESS FOR THE RB REGISTERS WHICH COMES FROM THE
                ;3163 ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;3164 ;  WITH A UNIQUE DATA PATTERN:
                ;3165 ;   R[0] 00000000
                ;3166 ;   R[1] 11111111
                ;3167 ;   R[2] 22222222
                ;3168 ;    ...
                ;3169 ;   R[0F] FFFFFFFF
                ;3170 ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;3171 ;  USING THE IB SP2.
                ;3172 ;
                ;3173 ; FPA UCODE USED
                ;3174 ;  LOCATION CONTENTS
                ;3175 ;  13:  BSC/R,SCR/R.R,FADC/LD,NAD/82
                ;3176 ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;3177 ;
                ;3178 ; LOGIC DESCRIPTION
                ;3179 ;
                ;3180 ; ERROR DESCRIPTION
                ;3181 ;  EXPECTED DATA READ FROM REGISTER SP2
                ;3182 ;  GOT DATA READ FROM REGISTER SP2
                ;3183 ;  SP2 REGISTER ADDRESS
                ;3184 ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP2
                ;3185 ;
                ;3186 ; SYNC POINT DESCRIPTION
                ;3187 ;
                ;3188 ;--
                ;3189 ;*****************************************************************
                ;3190 =0
U 10EC, 0000.003D.0180.0800.0000.108A  ;3191 RABTB: NEWTST
U 10ED, 0018.0038.1180.09F8.0000.10EE  ;3192  RC[0F]_K[.4]
U 10EE, 0018.0039.6180.09E0.0000.116E  ;3193 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;3194      ; COUNTER AND PUT THE DATA INTO
                ;3195      ; THE REGISTERS.
U 10EF, 0000.003C.1180.0800.0084.739C  ;3196  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 139C, 0818.0038.79F8.0800.0000.139D  ;3197  D_K[.30],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 139D, 0D00.003C.0180.0800.0000.139E  ;3198  D_DAL.SC
U 139E, 0001.003C.3180.09D8.0000.139F  ;3199  RC[0B]_D,KMX/.40 ; RC[0B]=300
U 139F, 0819.0014.3180.0980.0000.13A0  ;3200  ALU_D+K[.40],D_ALU,RC[0]_ALU ; RC[0]=340
U 13A0, 0019.0014.3180.0988.0000.13A1  ;3201  ALU_D+K[.40],RC[1]_ALU ; RC[1]=380
U 13A1, 0818.0038.6580.0800.0000.10F0  ;3202  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 10F0, 0819.0015.0D80.0800.0000.115E  ;3203 =0 ALU_D+K[.3],D_ALU,CALL,J/GENTRA
U 10F1, 0001.003C.0180.0990.0000.10F2  ;3204  RC[2]_D
U 10F2, 0000.003D.0180.0800.0000.114E  ;3205 =0 ERLOOP
U 10F3, 0000.003C.0180.0800.0000.10F4  ;3206  NOP
                ;3207 =0
U 10F4, 0000.003D.0180.0800.0000.1159  ;3208 RABTB1: CALL,J/FPINIT
```

```
U 10F5, 0010,0038,0180,0908,0200,10F6  ;3209  VA_RC[1]  ; LOAD THE PRN REGISTER WITH
U 10F6, 0000,003D,0180,0800,0000,1164  ;3210 =0 CALL,J/LOADIB  ; ADDRESS OF
U 10F7, 0000,003C,0180,0800,0000,1014  ;3211  NOP
U 1014, 0000,003D,0180,0800,0000,1015  ;3212 =00 CALL
U 1015, 0000,003F,0180,0800,0000,1200  ;3213  SUB/SPEC,J/IRTAB
U 1016, 0000,003C,0180,0800,0000,1017  ;3214  NOP
U 1017, 0010,0038,0180,0958,0200,10F8  ;3215  VA_RC[0B]  ; LOAD THE OPCODE WITH THE
U 10F8, 0000,003D,0180,0800,0000,1164  ;3216 =0 CALL,J/LOADIB  ; SP1 BEING TESTED INTO THE IB.
U 10F9, 0810,0038,0180,0910,0000,13A2  ;3217  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 13A2, 0000,003C,5980,3C00,0000,13A3  ;3218  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 13A3, 0000,00FC,0380,0800,0000,13A4  ;3219  TRAP.FPA
U 13A4, 0000,003C,0180,0A78,0000,13A5  ;3220  LAB_R[0F]  ; TEST INSTRUCTION IN FPA NOW.
U 13A5, 0000,003C,01D8,0800,0000,13A6  ;3221  Q_ACC  ; GET THE RESULT.
U 13A6, 0010,0038,0180,0960,0082,13A7  ;3222  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 13A7, 0800,003C,0180,0868,0000,13A8  ;3223  ALU_R(SC),D_ALU
U 13A8, 0001,003C,0180,09F0,0000,13A9  ;3224  RC[0E]_D
U 13A9, 0001,203C,0180,09E8,0000,13AA  ;3225  RC[0D]_Q
U 13AA, 001D,0000,0180,0800,0010,13AB  ;3226  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 13AB, 0000,013C,0180,0800,0000,10FA  ;3227  Z?
U 10FA, 0000,003D,0180,0800,0000,114F  ;3228 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 10FB, 0810,0038,0180,0958,0000,13AC  ;3229  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP2.
U 13AC, 0019,0014,1180,09D8,0000,13AD  ;3230  ALU_D+K[.4],RC[0B]_ALU
U 13AD, 0810,0038,0180,0960,0010,13AE  ;3231  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 13AE, 0000,013C,0180,0800,0000,10FC  ;3232  Z?
U 10FC, 0019,0000,0580,09E0,0000,10F4  ;3233 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTB1 ; NO SO CONTINUE.
U 10FD, 0000,003C,0180,0800,0000,10FE  ;3234  NOP
```

```
                ;3235  .PAGE "TEST 234 SP2+1 ON RA, GENERAL REGISTERS ADDRESSING TEST 3"
                ;3236 ;*****************************************************************
                ;3237 ;**
                ;3238 ; SP2+1 ON RA, GENERAL REGISTERS ADDRESSING TEST 3
                ;3239 ;
                ;3240 ; TEST DESCRIPTION
                ;3241 ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;3242 ;  SET COPY IN THE FPA. TESTED HERE IS THE (SPECIFIER 2)+1
                ;3243 ;  ADDRESS FOR THE RA REGISTERS WHICH COMES FROM THE
                ;3244 ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;3245 ;  WITH A UNIQUE DATA PATTERN:
                ;3246 ;   R[0] 00000000
                ;3247 ;   R[1] 11111111
                ;3248 ;   R[2] 22222222
                ;3249 ;    ...
                ;3250 ;   R[0F] FFFFFFFF
                ;3251 ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;3252 ;  USING THE IB SP2+1.
                ;3253 ;
                ;3254 ; FPA UCODE USED
                ;3255 ;  LOCATION CONTENTS
                ;3256 ;  12:  BSC/R,SCR/DPR.R,FADC/LD,NAD/82
                ;3257 ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;3258 ;
                ;3259 ; LOGIC DESCRIPTION
                ;3260 ;
                ;3261 ; ERROR DESCRIPTION
                ;3262 ;  EXPECTED DATA READ FROM REGISTER SP2+1
                ;3263 ;  GOT DATA READ FROM REGISTER SP2+1
                ;3264 ;  SP2 REGISTER ADDRESS
                ;3265 ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP2
                ;3266 ;
                ;3267 ; SYNC POINT DESCRIPTION
                ;3268 ;
                ;3269 ;--
                ;3270 ;*****************************************************************
                ;3271 =0
U 10FE, 0000.003D.0180.0800.0000.108A  ;3272 RABTC: NEWTST
U 10FF, 0018.0038.1180.09F8.0000.110A  ;3273  RC[0F]_K[.4]
U 110A, 0018.0039.6180.09E0.0000.116E  ;3274 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;3275      ; COUNTER AND PUT THE DATA INTO
                ;3276      ; THE REGISTERS.
U 110B, 0000.003C.1180.0800.0084.73AF  ;3277  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 13AF, 0818.0038.79F8.0800.0000.13B0  ;3278  D_K[.30],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 13B0, 0D00.003C.0180.0800.0000.13B1  ;3279  D_DAL.SC
U 13B1, 0001.003C.3180.09D8.0000.13B2  ;3280  RC[0B]_D.KMX/.40 ; RC[0B]=300
U 13B2, 0819.0014.3180.0980.0000.13B3  ;3281  ALU_D+K[.40],D_ALU,RC[0]_ALU ; RC[0]=340
U 13B3, 0019.0014.3180.0988.0000.13B4  ;3282  ALU_D+K[.40],RC[1]_ALU ; RC[1]=380
U 13B4, 0818.0038.6580.0800.0000.1110  ;3283  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 1110, 0819.0015.0980.0800.0000.115E  ;3284 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA
U 1111, 0001.003C.0180.0990.0000.1112  ;3285  RC[2]_D
U 1112, 0000.003D.0180.0800.0000.114E  ;3286 =0 ERLOOP
U 1113, 0000.003C.0180.0800.0000.1114  ;3287  NOP
                ;3288 =0
U 1114, 0000.003D.0180.0800.0000.1159  ;3289 RABTC1: CALL,J/FPINIT
```

```
U 1115. 0010.0038.0180.0958.0200.1116  ;3290  VA_RC[0B]  ; LOAD THE PRN REGISTER WITH
U 1116. 0000.003D.0180.0800.0000.1164  ;3291 =0 CALL,J/LOADIB  ; ADDRESS OF
U 1117. 0000.003C.0180.0800.0000.1020  ;3292  NOP
U 1020. 0000.003D.0180.0800.0000.1021  ;3293 =00 CALL
U 1021. 0000.003F.0180.0800.0000.1200  ;3294  SUB/SPEC,J/IRTAB
U 1022. 0000.003C.0180.0800.0000.1023  ;3295  NOP
U 1023. 0010.0038.0180.0908.0200.1118  ;3296  VA_RC[1]  ; LOAD THE OPCODE WITH THE
U 1118. 0000.003D.0180.0800.0000.1164  ;3297 =0 CALL,J/LOADIB  ; SP1 BEING TESTED INTO THE IB.
U 1119. 0810.0038.0180.0910.0000.13B5  ;3298  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 13B5. 0000.003C.5980.3C00.0000.13B6  ;3299  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 13B6. 0000.00FC.0380.0800.0000.13B7  ;3300  TRAP.FPA
U 13B7. 0000.003C.01D8.0A78.0000.13B8  ;3301  Q_ACC.LAB_R[0F]  ; GET THE RESULT.
U 13B8. 0010.0038.0180.0960.0082.13B9  ;3302  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 13B9. 0000.003C.0580.0800.0084.93BA  ;3303  SC_SC+K[.1]
U 13BA. 0800.003C.0180.0868.0000.13BB  ;3304  ALU_R(SC),D_ALU
U 13BB. 0001.003C.0180.09F0.0000.13BC  ;3305  RC[0E]_D
U 13BC. 0001.203C.0180.09E8.0000.13BD  ;3306  RC[0D]_Q
U 13BD. 001D.0000.0180.0800.0010.13BE  ;3307  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 13BE. 0000.013C.0180.0800.0000.111A  ;3308  Z?
U 111A. 0000.003D.0180.0800.0000.114F  ;3309 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 111B. 0810.0038.0180.0958.0000.13BF  ;3310  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP1.
U 13BF. 0019.0014.1180.09D8.0000.13C0  ;3311  ALU_D+K[.4],RC[0B]_ALU
U 13C0. 0810.0038.0180.0960.0010.13C1  ;3312  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 13C1. 0000.013C.0180.0800.0000.111C  ;3313  Z?
U 111C. 0019.0000.0580.09E0.0000.1114  ;3314 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTC1 ; NO SO CONTINUE.
U 111D. 0000.003C.0180.0800.0000.111E  ;3315  NOP
```

```
                ;3316  .PAGE "TEST 235 PRN+1 ON RB, GENERAL REGISTERS ADDRESSING TEST 4"
                ;3317  ;*****************************************************************
                ;3318  ;+
                ;3319  ; PRN+1 ON RB, GENERAL REGISTERS ADDRESSING TEST 4
                ;3320  ;
                ;3321  ; TEST DESCRIPTION
                ;3322  ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;3323  ;  SET COPY IN THE FPA. TESTED HERE IS THE PRN+1
                ;3324  ;  ADDRESS FOR THE RB REGISTERS WHICH COMES FROM THE
                ;3325  ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;3326  ;  WITH A UNIQUE DATA PATTERN:
                ;3327  ;   R[0] 00000000
                ;3328  ;   R[1] 11111111
                ;3329  ;   R[2] 22222222
                ;3330  ;    ...
                ;3331  ;   R[0F] FFFFFFFF
                ;3332  ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;3333  ;  USING THE IB PRN+1.
                ;3334  ;
                ;3335  ; FPA UCODE USED
                ;3336  ;  LOCATION CONTENTS
                ;3337  ;  12:  BSC/R,SCR/DPR.R,FADC/LD,NAD/82
                ;3338  ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;3339  ;
                ;3340  ; LOGIC DESCRIPTION
                ;3341  ;
                ;3342  ; ERROR DESCRIPTION
                ;3343  ;  EXPECTED DATA READ FROM REGISTER PRN+1
                ;3344  ;  GOT DATA READ FROM REGISTER PRN+1
                ;3345  ;  SP1 REGISTER ADDRESS
                ;3346  ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP1
                ;3347  ;
                ;3348  ; SYNC POINT DESCRIPTION
                ;3349  ;
                ;3350  ;--
                ;3351  ;*****************************************************************
                ;3352  =0
U 111E, 0000,003D,0180,0800,0000,108A  ;3353 RABTD: NEWTST
U 111F, 0018,0038,1180,09F8,0000,1120  ;3354  RC[0F]_K[.4]
U 1120, 0018,0039,6180,09E0,0000,116E  ;3355 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;3356      ; COUNTER AND PUT THE DATA INTO
                ;3357      ; THE REGISTERS.
U 1121, 0000,003C,1180,0800,0084,73C2  ;3358  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 13C2, 0818,0038,79F8,0800,0000,13C3  ;3359  D_K[.30],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 13C3, 0D00,003C,0180,0800,0000,13C4  ;3360  D_DAL.SC
U 13C4, 0001,003C,3180,0980,0000,13C5  ;3361  RC[0]_D,KMX/.40  ; RC[0]=300
U 13C5, 0819,0014,3180,09D8,0000,13C6  ;3362  ALU_D+K[.40],D_ALU,RC[0B]_ALU ; RC[0B]=340
U 13C6, 0019,0014,3180,0988,0000,13C7  ;3363  ALU_D+K[.40],RC[1]_ALU ; RC[1]=380
U 13C7, 0818,0038,6580,0800,0000,1122  ;3364  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 1122, 0819,0015,0980,0800,0000,115E  ;3365 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA
U 1123, 0001,003C,0180,0990,0000,1124  ;3366  RC[2]_D
U 1124, 0000,003D,0180,0800,0000,114E  ;3367 =0 ERLOOP
U 1125, 0000,003C,0180,0800,0000,1126  ;3368  NOP
                ;3369 =0
U 1126, 0000,003D,0180,0800,0000,1159  ;3370 RABTD1: CALL,J/FPINIT
```

```
U 1127. 0010.0038.0180.0958.0200.1128  ;3371  VA_RC[0B]  ; LOAD THE PRN REGISTER WITH
U 1128. 0000.003D.0180.0800.0000.1164  ;3372 =0 CALL,J/LOADIB  ; TEST ADDRESS
U 1129. 0000.003C.0180.0800.0000.1024  ;3373  NOP
U 1024. 0000.003D.0180.0800.0000.1025  ;3374 =00 CALL
U 1025. 0000.003F.0180.0800.0000.1200  ;3375  SUB/SPEC,J/IRTAB
U 1026. 0000.003C.0180.0800.0000.1027  ;3376  NOP
U 1027. 0010.0038.0180.0908.0200.112A  ;3377  VA_RC[1]  ; LOAD THE OPCODE WITH THE
U 112A. 0000.003D.0180.0800.0000.1164  ;3378 =0 CALL,J/LOADIB  ; SP1=0F INTO THE IB.
U 112B. 0810.0038.0180.0910.0000.13C8  ;3379  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 13C8. 0000.003C.5980.3C00.0000.13C9  ;3380  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 13C9. 0000.00FC.0380.0800.0000.13CA  ;3381  TRAP.FPA
U 13CA. 0000.003C.0180.0A78.0000.13CB  ;3382  LAB_R[0F]  ; TEST INSTRUCTION IN FPA
U 13CB. 0000.003C.01D8.0800.0000.13CC  ;3383  Q_ACC  ; GET THE RESULT.
U 13CC. 0010.0038.0180.0960.00B2.13CD  ;3384  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 13CD. 0000.003C.0580.0800.00B4.93CE  ;3385  SC_SC+K[.1]
U 13CE. 0800.003C.0180.0868.0000.13CF  ;3386  ALU_R(SC),D_ALU
U 13CF. 0001.003C.0180.09F0.0000.13D0  ;3387  RC[0E]_D
U 13D0. 0001.203C.0180.09E8.0000.13D1  ;3388  RC[0D]_Q
U 13D1. 001D.0000.0180.0800.0010.13D2  ;3389  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 13D2. 0000.013C.0180.0800.0000.112C  ;3390  Z?
U 112C. 0000.003D.0180.0800.0000.114F  ;3391 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 112D. 0810.0038.0180.0958.0000.13D3  ;3392  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP1.
U 13D3. 0019.0014.1180.09D8.0000.13D4  ;3393  ALU_D+K[.4],RC[0B]_ALU
U 13D4. 0810.0038.0180.0960.0010.13D5  ;3394  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 13D5. 0000.013C.0180.0800.0000.112E  ;3395  Z?
U 112E. 0019.0000.0580.09E0.0000.1126  ;3396 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTD1 ; NO SO CONTINUE.
U 112F. 0000.003C.0180.0800.0000.1130  ;3397  NOP
```

```
                ;3398  .PAGE "TEST 236 PRN*1 ON RA, GENERAL REGISTERS ADDRESSING TEST 5"
                ;3399  ;**********************************************************************
                ;3400  ;**
                ;3401  ; PRN*1 ON RA, GENERAL REGISTERS ADDRESSING TEST 5
                ;3402  ;
                ;3403  ; TEST DESCRIPTION
                ;3404  ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;3405  ;  SET COPY IN THE FPA. TESTED HERE IS THE PRN*1
                ;3406  ;  ADDRESS FOR THE RA REGISTERS WHICH COMES FROM THE
                ;3407  ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;3408  ;  WITH A UNIQUE DATA PATTERN:
                ;3409  ;   R[0] 00000000
                ;3410  ;   R[1] 11111111
                ;3411  ;   R[2] 22222222
                ;3412  ;    ...
                ;3413  ;   R[0F] FFFFFFFF
                ;3414  ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;3415  ;  USING THE IB PRN*1.
                ;3416  ;
                ;3417  ; FPA UCODE USED
                ;3418  ;  LOCATION CONTENTS
                ;3419  ;  14:  BSC/R,SCR/DP,FADC/LD,NAD/82
                ;3420  ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;3421  ;
                ;3422  ; LOGIC DESCRIPTION
                ;3423  ;
                ;3424  ; ERROR DESCRIPTION
                ;3425  ;  EXPECTED DATA READ FROM REGISTER PRN*1
                ;3426  ;  GOT DATA READ FROM REGISTER PRN*1
                ;3427  ;  SP1 REGISTER ADDRESS
                ;3428  ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP1
                ;3429  ;
                ;3430  ; SYNC POINT DESCRIPTION
                ;3431  ;
                ;3432  ;--
                ;3433  ;**********************************************************************
                ;3434  =0
U 1130.  0000.003D.0180.0800.0000.108A  ;3435 RABTE: NEWTST
U 1131.  0018.0038.1180.09F8.0000.1132  ;3436  RC[0F]_K[.4]
U 1132.  0018.0039.6180.09E0.0000.116E  ;3437 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;3438      ; COUNTER AND PUT THE DATA INTO
                ;3439      ; THE REGISTERS.
U 1133.  0000.003C.1180.0800.0084.73D6  ;3440  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 13D6.  0818.0038.79F8.0800.0000.13D7  ;3441  D_K[.30],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 13D7.  0D00.003C.0180.0800.0000.13D8  ;3442  D_DAL.SC
U 13D8.  0001.003C.3180.0980.0000.13D9  ;3443  RC[0]_D,KMX/.40  ; RC[0]=300
U 13D9.  0819.0014.3180.09D8.0000.13DA  ;3444  ALU_D*K[.40],D_ALU,RC[0B]_ALU ; RC[0B]=340
U 13DA.  0019.0014.3180.0988.0000.13DB  ;3445  ALU_D*K[.40],RC[1]_ALU ; RC[1]=380
U 13DB.  0818.0038.6580.0800.0000.1134  ;3446  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 1134.  0819.0015.1180.0800.0000.115E  ;3447 =0 ALU_D*K[.4],D_ALU,CALL,J/GENTRA
U 1135.  0001.003C.0180.0990.0000.1136  ;3448  RC[2]_D
U 1136.  0000.003D.0180.0800.0000.114E  ;3449 =0 ERLOOP
U 1137.  0000.003C.0180.0800.0000.1138  ;3450  NOP
                ;3451 =0
U 1138.  0000.003D.0180.0800.0000.1159  ;3452 RABTE1: CALL,J/FPINIT
```

```
U 1139, 0010,0038,0180,0958,0200,113A  ;3453  VA_RC[0B]  ; LOAD THE PRN REGISTER WITH
U 113A, 0000,003D,0180,0800,0000,1164  ;3454 =0 CALL,J/LOADIB  ; TEST ADDRESS
U 113B, 0000,003C,0180,0800,0000,1030  ;3455  NOP
U 1030, 0000,003D,0180,0800,0000,1031  ;3456 =00 CALL
U 1031, 0000,003F,0180,0800,0000,1200  ;3457  SUB/SPEC,J/IRTAB
U 1032, 0000,003C,0180,0800,0000,1033  ;3458  NOP
U 1033, 0010,0038,0180,0908,0200,113C  ;3459  VA_RC[1]  ; LOAD THE OPCODE WITH THE
U 113C, 0000,003D,0180,0800,0000,1164  ;3460 =0 CALL,J/LOADIB  ; SP1=0F INTO THE IB.
U 113D, 0810,0038,0180,0910,0000,13DC  ;3461  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 13DC, 0000,003C,5980,3C00,0000,13DD  ;3462  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 13DD, 0000,00FC,0380,0800,0000,13DE  ;3463  TRAP.FPA
U 13DE, 0000,003C,01D8,0A78,0000,13DF  ;3464  Q_ACC,LAB_R[0F]  ; GET THE RESULT.
U 13DF, 0010,0038,0180,0960,0082,13E0  ;3465  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 13E0, 0000,003C,0580,0800,0084,93E1  ;3466  SC_SC+K[.1]
U 13E1, 0800,003C,0180,0868,0000,13E2  ;3467  ALU_R(SC),D_ALU
U 13E2, 0001,003C,0180,09F0,0000,13E3  ;3468  RC[0E]_D
U 13E3, 0001,203C,0180,09E8,0000,13E4  ;3469  RC[0D]_Q
U 13E4, 001D,0000,0180,0800,0010,13E5  ;3470  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 13E5, 0000,013C,0180,0800,0000,113E  ;3471  Z?
U 113E, 0000,003D,0180,0800,0000,114F  ;3472 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 113F, 0810,0038,0180,0958,0000,13E6  ;3473  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP1.
U 13E6, 0019,0014,1180,09D8,0000,13E7  ;3474  ALU_D+K[.4],RC[0B]_ALU
U 13E7, 0810,0038,0180,0960,0010,13E8  ;3475  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 13E8, 0000,013C,0180,0800,0000,1140  ;3476  Z?
U 1140, 0019,0000,0580,09E0,0000,1138  ;3477 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTE1 ; NO SO CONTINUE.
U 1141, 0000,003C,0180,0800,0000,1142  ;3478  NOP
          ;3479
```

```
          ;3480  .PAGE "TEST 237 RESERVED"
          ;3481 =0
U 1142. 0000.003D.0180.0800.0000.108A  ;3482 D1: NEWTST
U 1143. 0000.003C.0180.0800.0000.1144  ;3483  NOP
          ;3484
```

```
            ;3485  .PAGE "TEST 238 RESERVED"
            ;3486 =0
U 1144, 0000,003D,0180,0800,0000,108A  ;3487 D2: NEWTST
U 1145, 0000,003C,0180,0800,0000,1146  ;3488  NOP
            ;3489 =0
U 1146, 0000,003D,0180,0800,0000,108A  ;3490 D3: NEWTST
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 3019= 3020= 3021= 3022=
U 1008  1898= 1899= 1875 2338= 1918= 1919= 1876 2339=
U 1010  3081= 3082= 3083= 3084= 3212= 3213= 3214= 3215=
U 1018  1995= 1996= 1877 2345= 1998= 1999= 1878 2346=
U 1020  3293= 3294= 3295= 3296= 3374= 3375= 3376= 3377=
U 1028  2000= 2001= 1879 2355= 2003= 2004= 1880 2356=
U 1030  3456= 3457= 3458= 3459= 2023= 2024= 2330= 2331=
U 1038  2364= 2365= 1900 2360= 2451= 2452= 1901 2361=
U 1040  2455= 2456= 2467= 2468= 2470= 2471= 2491= 2492=
U 1048  2495= 2496= 1902 2373= 2502= 2503= 1939: 2374=
U 1050  2506= 2507= 2512= 2513= 2567= 2568= 2570= 2571=
U 1058  2578= 2579= 1942: 2609= 2580= 2581= 1945: 2610=
U 1060  2583= 2584= 2594= 2595= 2601= 2602= 2616= 2617=
U 1068  2618= 2619= 1903 2657= 2626= 2627= 1904 2658=
U 1070  2631= 2632= 2642= 2643= 2649= 2650= 2665= 2666=
U 1078  2667= 2668= 1905 2706= 2675= 2676= 1906 2707=
U 1080  2678= 2679= 2681= 2682= 2689= 2690= 2697= 2698=
U 1088  2713= 2714= 1916 2755= 2715= 2716= 1917 2756=
U 1090  2723= 2724= 2728= 2729= 2731= 2732= 2739= 2740=
U 1098  2747= 2748= 1920 2838= 2805= 2806= 1921 2839=
U 10A0  2809= 2810= 2814= 2815= 2817= 2818= 2827= 2828=
U 10A8  2832= 2833= 1922 2921= 2845= 2846= 1923 2922=
U 10B0  2890= 2891= 2892= 2893= 2897= 2898= 2900= 2901=
U 10B8  2910= 2911= 1924 2938= 2915= 2916= 1925 2939=
U 10C0  2928= 2929= 2930= 2931= 2943= 2944= 2981= 2982=
U 10C8  2984= 2985= 1926 3001= 2989= 2990= 1927 3002=
U 10D0  2992= 2993= 2997= 2998= 3005= 3006= 3013= 3014=
U 10D8  3017= 3018= 3060= 3061= 3062= 3065= 3072= 3073=
U 10E0  3074= 3075= 3077= 3078= 3079= 3080= 3085= 3086=
U 10E8  3096= 3097= 3101= 3102= 3191= 3192= 3193= 3196=
U 10F0  3203= 3204= 3205= 3206= 3208= 3209= 3210= 3211=
U 10F8  3216= 3217= 3228= 3229= 3233= 3234= 3272= 3273=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1928 3274= 3277= 1870: 1871: 1872: 1873:
U 1110  3284= 3285= 3286= 3287= 3289= 3290= 3291= 3292=
U 1118  3297= 3298= 3309= 3310= 3314= 3315= 3353= 3354=
U 1120  3355= 3358= 3365= 3366= 3367= 3368= 3370= 3371=
U 1128  3372= 3373= 3378= 3379= 3391= 3392= 3396= 3397=
U 1130  3435= 3436= 3437= 3440= 3447= 3448= 3449= 3450=
U 1138  3452= 3453= 3454= 3455= 3460= 3461= 3472= 3473=
U 1140  3477= 3478= 3482= 3483= 3487= 3488= 3490= 1929
U 1148  1930 1931 1932 1933 1934 1935 1936 1937
U 1150  1938 1940 1941 1973 1974 1965: 1980 1981
U 1158  1982 1993 1994 1997 2002 2005 2015 2016
U 1160  2017 2025 2026 2027 2033 2034 2035 2036
U 1168  2037 2038 2039 2040 2041 2042 2315 2316
U 1170  2317 2318 2319 2320 2321 2322 2323 2324
U 1178  2325 2326 2327 2328 2329 2334 2335 2336
U 1180  2337 2340 2341 2342 2343 2344 2348 2349
U 1188  2350 2351 2352 2353 2354 2357 2358 2359
U 1190  2362 2363 2366 2367 2368 2369 2370 2371
U 1198  2372 2375 2376 2377 2378 2379 2387 2388
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2389 2390 2391 2392 2393 2453 2454 2457
U 11A8  2458 2459 2460 2461 2462 2463 2464 2465
U 11B0  2466 2472 2473 2474 2475 2476 2477 2478
U 11B8  2479 2480 2481 2482 2483 2484 2485 2486
U 11C0  2487 2488 2489 2490 2493 2494 2497 2498
U 11C8  2499 2500 2501 2504 2505 2508 2509 2510
U 11D0  2511 2569 2572 2573 2574 2575 2576 2577
U 11D8  2585 2586 2587 2588 2589 2590 2591 2592
U 11E0  2593 2596 2597 2598 2599 2600 2603 2604
U 11E8  2605 2606 2607 2608 2620 2621 2622 2623
U 11F0  2624 2625 2628 2629 2630 2633 2634 2635
U 11F8  2636 2637 2638 2639 2640 2641 2644 2645
U 1200  2049: 2050: 2051: 2052: 2053: 2054: 2055: 2056:
U 1208  2057: 2058: 2059: 2060: 2061: 2062: 2063: 2064:
U 1210  2065: 2066: 2067: 2068: 2069: 2070: 2071: 2072:
U 1218  2073: 2074: 2075: 2076: 2077: 2078: 2079: 2080:
U 1220  2081: 2082: 2083: 2084: 2085: 2086: 2087: 2088:
U 1228  2089: 2090: 2091: 2092: 2093: 2094: 2095: 2096:
U 1230  2097: 2098: 2099: 2100: 2101: 2102: 2103: 2104:
U 1238  2105: 2106: 2107: 2108: 2109: 2110: 2111: 2112:
U 1240  2113: 2114: 2115: 2116: 2117: 2118: 2119: 2120:
U 1248  2121: 2122: 2123: 2124: 2125: 2126: 2127: 2128:
U 1250  2129: 2130: 2131: 2132: 2133: 2134: 2135: 2136:
U 1258  2137: 2138: 2139: 2140: 2141: 2142: 2143: 2144:
U 1260  2145: 2146: 2147: 2148: 2149: 2150: 2151: 2152:
U 1268  2153: 2154: 2155: 2156: 2157: 2158: 2159: 2160:
U 1270  2161: 2162: 2163: 2164: 2165: 2166: 2167: 2168:
U 1278  2169: 2170: 2171: 2172: 2173: 2174: 2175: 2176:
U 1280  2177: 2178: 2179: 2180: 2181: 2182: 2183: 2184:
U 1288  2185: 2186: 2187: 2188: 2189: 2190: 2191: 2192:
U 1290  2193: 2194: 2195: 2196: 2197: 2198: 2199: 2200:
U 1298  2201: 2202: 2203: 2204: 2205: 2206: 2207: 2208:
U 12A0  2209: 2210: 2211: 2212: 2213: 2214: 2215: 2216:
U 12A8  2217: 2218: 1966: 2219: 2220: 2221: 2222: 2223:
U 12B0  2224: 2225: 2226: 2227: 2228: 2229: 2230: 2231:
U 12B8  2232: 2233: 2234: 2235: 2236: 2237: 2238: 2239:
U 12C0  2240: 2241: 2242: 2243: 2244: 2245: 2246: 2247:
U 12C8  2248: 2249: 2250: 2251: 2252: 2253: 2254: 2255:
U 12D0  2256: 2257: 2258: 2259: 2260: 2261: 2262: 2263:
U 12D8  2264: 2265: 2266: 2267: 2268: 2269: 2270: 2271:
U 12E0  2272: 2273: 2274: 2275: 2276: 2277: 2278: 2279:
U 12E8  2280: 2281: 2282: 2283: 2284: 2285: 2286: 2287:
U 12F0  2288: 2289: 2290: 2291: 2292: 2293: 2294: 2295:
U 12F8  2296: 2297: 2298: 2299: 2300: 2301: 2302: 2303:
U 1300  2646 2647 2648 2651 2652 2653 2654 2655
U 1308  2656 2669 2670 2671 2672 2673 2674 2680
U 1310  2683 2684 2685 2686 2687 2688 2691 2692
U 1318  2693 2694 2695 2696 2699 2700 2701 2702
U 1320  2703 2704 2705 2717 2718 2719 2720 2721
U 1328  2722 2725 2726 2727 2730 2733 2734 2735
U 1330  2736 2737 2738 2741 2742 2743 2744 2745
U 1338  2746 2749 2750 2751 2752 2753 2754 2807
U 1340  2808 2811 2812 2813 2819 2820 2821 2822
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2386
ESKARSB.MCR                    MICRO2  1P(00)    26-JUL-85  17:14:26     Page   9

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  2823 2824 2825 2826 2829 2830 2831 2834
U 1350  2835 2836 2837 2840 2841 2842 2843 2844
U 1358  2894 2895 2896 2902 2903 2904 2905 2906
U 1360  2907 2908 2909 2912 2913 2914 2917 2918
U 1368  2919 2920 2923 2924 2925 2926 2927 2932
U 1370  2933 2934 2935 2936 2937 2940 2941 2942
U 1378  2983 2986 2987 2988 2994 2995 2996 2999
U 1380  3000 3003 3007 3008 3009 3010 3011 3012
U 1388  3015 3016 3066 3067 3068 3069 3070 3071
U 1390  3087 3088 3089 3090 3091 3092 3093 3094
U 1398  3095 3098 3099 3100 3197 3198 3199 3200
U 13A0  3201 3202 3218 3219 3220 3221 3222 3223
U 13A8  3224 3225 3226 3227 3230 3231 3232 3278
U 13B0  3279 3280 3281 3282 3283 3299 3300 3301
U 13B8  3302 3303 3304 3305 3306 3307 3308 3311
U 13C0  3312 3313 3359 3360 3361 3362 3363 3364
U 13C8  3380 3381 3382 3383 3384 3385 3386 3387
U 13D0  3388 3389 3390 3393 3394 3395 3441 3442
U 13D8  3443 3444 3445 3446 3462 3463 3464 3465
U 13E0  3466 3467 3468 3469 3470 3471 3474 3475
U 13E8  3476
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2387
ESKAR5B.MCR                    MICRO2  1P(00)    26-JUL-85  17:14:26     Page   9

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR5B       1017

Used               1017
Remaining

Total microwords used in memory U: 1017
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2388
ESKAR5B.MCR                    MICRO2  1P(00)    26-JUL-85  17:14:26           Page   9


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR5B.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR5C.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 2389

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2515 TEST 239 PRN ON RB, GENERAL REGISTERS ADDRESSING TEST 6
; 2649 TEST 23A CPSYNC/WAIT, CLEAR UWORD TEST
; 2722 TEST 23B LITERAL REGISTER TEST
; 2817 TEST 23C SCRATCH PAD DATA TYPE TEST
; 2984 TEST 23D CONFLICTING ADDRESS, SCRATCH PAD TEST
; 3158 TEST 23E SA AND SB SIGN LATCH TEST
; 3491 TEST 23F RESERVED
; 3496 TEST 240 RESERVED
; 3501 TEST 241 RESERVED
; 3506 TEST 242 RESERVED
; 3511 TEST 243 RESERVED
; 3516 TEST 244 RESERVED
; 3521 TEST 245 RESERVED
; 3526 TEST 246 RESERVED
; 3531 TEST 247 RESERVED
; 3536 TEST 248 RESERVED

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR5C
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.100A  ;1874 TRPH0: Q_ID[USTACK]
U 100A. 0C00.003C.01E0.0800.0000.100E  ;1875  Q_D,D_Q
U 100E. 0000.003C.8180.3C00.0000.1016  ;1876  ID[USTACK]_D
U 1016. 0C00.003C.01E0.0800.0000.1018  ;1877  Q_D,D_Q
U 1018. 0000.003C.01C0.0A78.0000.101E  ;1878  Q_R[0F]
U 101E. 0001.2024.0180.0800.0010.1026  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1026. 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1008  ;1897 TRPH1: RC[0D]_SC
U 1008, 0000.003D.D980.0800.0084.7166  ;1898 =0 SETRCF[.9]
U 1009, 0000.003C.49F0.2C00.0000.102E  ;1899  Q_ID[TBER0]
U 102E, 0001.203C.4DF0.2DB0.0000.1036  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1036, 0001.203C.65F0.2DB8.0000.103E  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 103E, 0001.203C.6DF0.2DD8.0000.1046  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1046, 0001.203C.75F0.2DE0.0000.1056  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1056, 0001.203C.79F0.2DC8.0000.105B  ;1904  RC[9]_Q,Q_ID[PARITY]
U 105B, 0001.203C.69F0.2DC0.0000.106A  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 106A, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1160  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 106E, 0000.003C.8580.0800.0084.7076  ;1916 SCOPE: SC_K[.C]
U 1076, 0803.001C.0180.0800.0000.100C  ;1917  ALU_NOT.MASK,D_ALU
U 100C, 0000.003D.7580.3C00.0000.1171  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 100D, 0000.003C.65F8.0800.0084.707E  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 107E, 0818.0038.8D80.0800.0000.1109  ;1920  D_K[.1F]    ; ...
U 1109, 0D00.003C.0180.0800.0000.1117  ;1921  D_DAL.SC
U 1117, 0000.003C.3D80.3C00.0000.1151  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 1151, 0818.0038.0580.0800.0000.1152  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 1152, 0D00.003C.0180.0800.0000.1153  ;1924  D_DAL.SC
U 1153, 0F00.003C.3180.3C00.0000.1154  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1154, 0000.003C.0180.0800.0000.1156  ;1926  NOP
U 1156, 0000.003C.3980.3C00.0000.1157  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1157, 0000.003C.2980.3C00.0000.1158  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1158, 0000.003C.4980.3C00.0000.1159  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1159, 0818.0038.C180.0800.0000.115A  ;1930  D_K[.FFFF]
U 115A, 0000.003C.6580.3C00.0000.115B  ;1931  ID[SBI.ERR]_D
U 115B, 0F00.003C.6D80.3C00.0000.115C  ;1932  ID[FAULT]_D,D_0
U 115C, 0000.003C.6D80.3C00.0000.115D  ;1933  ID[FAULT]_D
U 115D, 0000.003C.6580.3C00.0000.115E  ;1934  ID[SBI.ERR]_D
U 115E, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 115F, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 1160, 0810.0038.0180.0978.0000.1161  ;1937 ERROR1: D_RC[0F]
U 1161, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 1162, 0810.0038.0180.0978.0000.1163  ;1940 ERROR2: D_RC[0F]
U 1163, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2. 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1164  ;1964 13FF: NOP
          ;1965
          ;1966
          ;1967 ;*
          ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
          ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
          ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
          ;1971 ; TYPING IT.
          ;1972 ;-
U 1164. 0810.0038.4D80.0978.0000.1165  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1165. 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
          ;1975 ;*
          ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
          ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
          ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
          ;1979 ;-
U 1166. 0010.0038.01C0.0978.0000.1167  ;1980 SETRCF: Q_RC[0F]
U 1167. 0019.2034.4DC0.0800.0000.1168  ;1981  Q_Q.AND.K[.FF00]
U 1168. 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
          ;1983
          ;1984 ;
          ;1985 ; THIS TABLE IS USED TO CATCH THE IB CALL 3 BRANCH WHEN IT IS
          ;1986 ; NECESSARY FOR A CALL 3 TO BE USED IN FPA TESTING.
          ;1987 ;
          ;1988 1200:
          ;1989 IRTAB:
U 1200. 0001.203E.59F0.2DE8.0000.0002  ;1990 1200: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1201. 0001.203E.59F0.2DE8.0000.0002  ;1991 1201: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1202. 0001.203E.59F0.2DE8.0000.0002  ;1992 1202: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1203. 0001.203E.59F0.2DE8.0000.0002  ;1993 1203: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1204. 0001.203E.59F0.2DE8.0000.0002  ;1994 1204: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1205. 0001.203E.59F0.2DE8.0000.0002  ;1995 1205: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1206. 0001.203E.59F0.2DE8.0000.0002  ;1996 1206: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1207. 0001.203E.59F0.2DE8.0000.0002  ;1997 1207: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1208. 0001.203E.59F0.2DE8.0000.0002  ;1998 1208: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 1209. 0001.203E.59F0.2DE8.0000.0002  ;1999 1209: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120A. 0001.203E.59F0.2DE8.0000.0002  ;2000 120A: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120B. 0001.203E.59F0.2DE8.0000.0002  ;2001 120B: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120C. 0001.203E.59F0.2DE8.0000.0002  ;2002 120C: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120D. 0001.203E.59F0.2DE8.0000.0002  ;2003 120D: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120E. 0001.203E.59F0.2DE8.0000.0002  ;2004 120E: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 120F. 0001.203E.59F0.2DE8.0000.0002  ;2005 120F: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 1210.  0001.203E.59F0.2DE8.0000.0002  ;2006 1210: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1211.  0001.203E.59F0.2DE8.0000.0002  ;2007 1211: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1212.  0001.203E.59F0.2DE8.0000.0002  ;2008 1212: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1213.  0001.203E.59F0.2DE8.0000.0002  ;2009 1213: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1214.  0001.203E.59F0.2DE8.0000.0002  ;2010 1214: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1215.  0001.203E.59F0.2DE8.0000.0002  ;2011 1215: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1216.  0001.203E.59F0.2DE8.0000.0002  ;2012 1216: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1217.  0001.203E.59F0.2DE8.0000.0002  ;2013 1217: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1218.  0001.203E.59F0.2DE8.0000.0002  ;2014 1218: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1219.  0001.203E.59F0.2DE8.0000.0002  ;2015 1219: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121A.  0001.203E.59F0.2DE8.0000.0002  ;2016 121A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121B.  0001.203E.59F0.2DE8.0000.0002  ;2017 121B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121C.  0001.203E.59F0.2DE8.0000.0002  ;2018 121C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121D.  0001.203E.59F0.2DE8.0000.0002  ;2019 121D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121E.  0001.203E.59F0.2DE8.0000.0002  ;2020 121E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 121F.  0001.203E.59F0.2DE8.0000.0002  ;2021 121F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1220.  0001.203E.59F0.2DE8.0000.0002  ;2022 1220: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1221.  0001.203E.59F0.2DE8.0000.0002  ;2023 1221: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1222.  0001.203E.59F0.2DE8.0000.0002  ;2024 1222: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1223.  0001.203E.59F0.2DE8.0000.0002  ;2025 1223: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1224.  0001.203E.59F0.2DE8.0000.0002  ;2026 1224: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1225.  0001.203E.59F0.2DE8.0000.0002  ;2027 1225: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1226.  0001.203E.59F0.2DE8.0000.0002  ;2028 1226: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1227.  0001.203E.59F0.2DE8.0000.0002  ;2029 1227: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1228.  0001.203E.59F0.2DE8.0000.0002  ;2030 1228: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1229.  0001.203E.59F0.2DE8.0000.0002  ;2031 1229: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122A.  0001.203E.59F0.2DE8.0000.0002  ;2032 122A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122B.  0001.203E.59F0.2DE8.0000.0002  ;2033 122B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122C.  0001.203E.59F0.2DE8.0000.0002  ;2034 122C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122D.  0001.203E.59F0.2DE8.0000.0002  ;2035 122D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122E.  0001.203E.59F0.2DE8.0000.0002  ;2036 122E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 122F.  0001.203E.59F0.2DE8.0000.0002  ;2037 122F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1230.  0001.203E.59F0.2DE8.0000.0002  ;2038 1230: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1231.  0001.203E.59F0.2DE8.0000.0002  ;2039 1231: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1232.  0001.203E.59F0.2DE8.0000.0002  ;2040 1232: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1233.  0001.203E.59F0.2DE8.0000.0002  ;2041 1233: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1234.  0001.203E.59F0.2DE8.0000.0002  ;2042 1234: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1235.  0001.203E.59F0.2DE8.0000.0002  ;2043 1235: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1236.  0001.203E.59F0.2DE8.0000.0002  ;2044 1236: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1237.  0001.203E.59F0.2DE8.0000.0002  ;2045 1237: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1238.  0001.203E.59F0.2DE8.0000.0002  ;2046 1238: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1239.  0001.203E.59F0.2DE8.0000.0002  ;2047 1239: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123A.  0001.203E.59F0.2DE8.0000.0002  ;2048 123A: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123B.  0001.203E.59F0.2DE8.0000.0002  ;2049 123B: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123C.  0001.203E.59F0.2DE8.0000.0002  ;2050 123C: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123D.  0001.203E.59F0.2DE8.0000.0002  ;2051 123D: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123E.  0001.203E.59F0.2DE8.0000.0002  ;2052 123E: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 123F.  0001.203E.59F0.2DE8.0000.0002  ;2053 123F: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1240.  0001.203E.59F0.2DE8.0000.0002  ;2054 1240: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1241.  0001.203E.59F0.2DE8.0000.0002  ;2055 1241: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1242.  0001.203E.59F0.2DE8.0000.0002  ;2056 1242: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1243.  0001.203E.59F0.2DE8.0000.0002  ;2057 1243: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1244.  0001.203E.59F0.2DE8.0000.0002  ;2058 1244: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1245.  0001.203E.59F0.2DE8.0000.0002  ;2059 1245: RC[0D]_Q.Q_ID[ACC.2].RETURN2
U 1246.  0001.203E.59F0.2DE8.0000.0002  ;2060 1246: RC[0D]_Q.Q_ID[ACC.2].RETURN2
```

```
U 1247.  0001.203E.59F0.2DE8.0000.0002  ;2061 1247: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1248.  0001.203E.59F0.2DE8.0000.0002  ;2062 1248: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1249.  0001.203E.59F0.2DE8.0000.0002  ;2063 1249: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124A.  0001.203E.59F0.2DE8.0000.0002  ;2064 124A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124B.  0001.203E.59F0.2DE8.0000.0002  ;2065 124B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124C.  0001.203E.59F0.2DE8.0000.0002  ;2066 124C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124D.  0001.203E.59F0.2DE8.0000.0002  ;2067 124D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124E.  0001.203E.59F0.2DE8.0000.0002  ;2068 124E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 124F.  0001.203E.59F0.2DE8.0000.0002  ;2069 124F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1250.  0001.203E.59F0.2DE8.0000.0002  ;2070 1250: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1251.  0001.203E.59F0.2DE8.0000.0002  ;2071 1251: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1252.  0001.203E.59F0.2DE8.0000.0002  ;2072 1252: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1253.  0001.203E.59F0.2DE8.0000.0002  ;2073 1253: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1254.  0001.203E.59F0.2DE8.0000.0002  ;2074 1254: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1255.  0001.203E.59F0.2DE8.0000.0002  ;2075 1255: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1256.  0001.203E.59F0.2DE8.0000.0002  ;2076 1256: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1257.  0001.203E.59F0.2DE8.0000.0002  ;2077 1257: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1258.  0001.203E.59F0.2DE8.0000.0002  ;2078 1258: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1259.  0001.203F.59F0.2DE8.0000.0002  ;2079 1259: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125A.  0001.203E.59F0.2DE8.0000.0002  ;2080 125A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125B.  0001.203E.59F0.2DE8.0000.0002  ;2081 125B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125C.  0001.203E.59F0.2DE8.0000.0002  ;2082 125C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125D.  0001.203E.59F0.2DE8.0000.0002  ;2083 125D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125E.  0001.203E.59F0.2DE8.0000.0002  ;2084 125E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 125F.  0001.203E.59F0.2DE8.0000.0002  ;2085 125F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1260.  0001.203E.59F0.2DE8.0000.0002  ;2086 1260: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1261.  0001.203E.59F0.2DE8.0000.0002  ;2087 1261: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1262.  0001.203E.59F0.2DE8.0000.0002  ;2088 1262: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1263.  0001.203E.59F0.2DE8.0000.0002  ;2089 1263: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1264.  0001.203E.59F0.2DE8.0000.0002  ;2090 1264: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1265.  0001.203E.59F0.2DE8.0000.0002  ;2091 1265: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1266.  0001.203E.59F0.2DE8.0000.0002  ;2092 1266: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1267.  0001.203E.59F0.2DE8.0000.0002  ;2093 1267: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1268.  0001.203E.59F0.2DE8.0000.0002  ;2094 1268: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1269.  0001.203E.59F0.2DE8.0000.0002  ;2095 1269: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126A.  0001.203E.59F0.2DE8.0000.0002  ;2096 126A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126B.  0001.203E.59F0.2DE8.0000.0002  ;2097 126B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126C.  0001.203E.59F0.2DE8.0000.0002  ;2098 126C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126D.  0001.203E.59F0.2DE8.0000.0002  ;2099 126D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126E.  0001.203E.59F0.2DE8.0000.0002  ;2100 126E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 126F.  0001.203E.59F0.2DE8.0000.0002  ;2101 126F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1270.  0001.203E.59F0.2DE8.0000.0002  ;2102 1270: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1271.  0001.203E.59F0.2DE8.0000.0002  ;2103 1271: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1272.  0001.203E.59F0.2DE8.0000.0002  ;2104 1272: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1273.  0001.203E.59F0.2DE8.0000.0002  ;2105 1273: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1274.  0001.203E.59F0.2DE8.0000.0002  ;2106 1274: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1275.  0001.203E.59F0.2DE8.0000.0002  ;2107 1275: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1276.  0001.203E.59F0.2DE8.0000.0002  ;2108 1276: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1277.  0001.203E.59F0.2DE8.0000.0002  ;2109 1277: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1278.  0001.203E.59F0.2DE8.0000.0002  ;2110 1278: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1279.  0001.203E.59F0.2DE8.0000.0002  ;2111 1279: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 127A.  0001.203E.59F0.2DE8.0000.0002  ;2112 127A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 127B.  0001.203E.59F0.2DE8.0000.0002  ;2113 127B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 127C.  0001.203E.59F0.2DE8.0000.0002  ;2114 127C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 127D.  0001.203E.59F0.2DE8.0000.0002  ;2115 127D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 2398
ESKARSC.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14                    Page    5

U 127E. 0001.203E.59F0.2DE8.0000.0002  ;2116 127E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 127F. 0001.203E.59F0.2DE8.0000.0002  ;2117 127F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1280. 0001.203E.59F0.2DE8.0000.0002  ;2118 1280: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1281. 0001.203E.59F0.2DE8.0000.0002  ;2119 1281: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1282. 0001.203E.59F0.2DE8.0000.0002  ;2120 1282: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1283. 0001.203E.59F0.2DE8.0000.0002  ;2121 1283: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1284. 0001.203E.59F0.2DE8.0000.0002  ;2122 1284: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1285. 0001.203E.59F0.2DE8.0000.0002  ;2123 1285: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1286. 0001.203E.59F0.2DE8.0000.0002  ;2124 1286: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1287. 0001.203E.59F0.2DE8.0000.0002  ;2125 1287: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1288. 0001.203E.59F0.2DE8.0000.0002  ;2126 1288: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1289. 0001.203E.59F0.2DE8.0000.0002  ;2127 1289: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128A. 0001.203E.59F0.2DE8.0000.0002  ;2128 128A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128B. 0001.203E.59F0.2DE8.0000.0002  ;2129 128B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128C. 0001.203E.59F0.2DE8.0000.0002  ;2130 128C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128D. 0001.203E.59F0.2DE8.0000.0002  ;2131 128D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128E. 0001.203E.59F0.2DE8.0000.0002  ;2132 128E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 128F. 0001.203E.59F0.2DE8.0000.0002  ;2133 128F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1290. 0001.203E.59F0.2DE8.0000.0002  ;2134 1290: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1291. 0001.203E.59F0.2DE8.0000.0002  ;2135 1291: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1292. 0001.203E.59F0.2DE8.0000.0002  ;2136 1292: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1293. 0001.203E.59F0.2DE8.0000.0002  ;2137 1293: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1294. 0001.203E.59F0.2DE8.0000.0002  ;2138 1294: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1295. 0001.203E.59F0.2DE8.0000.0002  ;2139 1295: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1296. 0001.203E.59F0.2DE8.0000.0002  ;2140 1296: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1297. 0001.203E.59F0.2DE8.0000.0002  ;2141 1297: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1298. 0001.203E.59F0.2DE8.0000.0002  ;2142 1298: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 1299. 0001.203E.59F0.2DE8.0000.0002  ;2143 1299: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129A. 0001.203E.59F0.2DE8.0000.0002  ;2144 129A: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129B. 0001.203E.59F0.2DE8.0000.0002  ;2145 129B: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129C. 0001.203E.59F0.2DE8.0000.0002  ;2146 129C: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129D. 0001.203E.59F0.2DE8.0000.0002  ;2147 129D: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129E. 0001.203E.59F0.2DE8.0000.0002  ;2148 129E: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 129F. 0001.203E.59F0.2DE8.0000.0002  ;2149 129F: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A0. 0001.203E.59F0.2DE8.0000.0002  ;2150 12A0: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A1. 0001.203E.59F0.2DE8.0000.0002  ;2151 12A1: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A2. 0001.203E.59F0.2DE8.0000.0002  ;2152 12A2: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A3. 0001.203E.59F0.2DE8.0000.0002  ;2153 12A3: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A4. 0001.203E.59F0.2DE8.0000.0002  ;2154 12A4: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A5. 0001.203E.59F0.2DE8.0000.0002  ;2155 12A5: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A6. 0001.203E.59F0.2DE8.0000.0002  ;2156 12A6: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A7. 0001.203E.59F0.2DE8.0000.0002  ;2157 12A7: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A8. 0001.203E.59F0.2DE8.0000.0002  ;2158 12A8: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12A9. 0001.203E.59F0.2DE8.0000.0002  ;2159 12A9: RC[0D]_Q.Q_ID[ACC.2],RETURN2
          ;2160
U 12AB. 0001.203E.59F0.2DE8.0000.0002  ;2161 12AB: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12AC. 0001.203E.59F0.2DE8.0000.0002  ;2162 12AC: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12AD. 0001.203E.59F0.2DE8.0000.0002  ;2163 12AD: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12AE. 0001.203E.59F0.2DE8.0000.0002  ;2164 12AE: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12AF. 0001.203E.59F0.2DE8.0000.0002  ;2165 12AF: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12B0. 0001.203E.59F0.2DE8.0000.0002  ;2166 12B0: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12B1. 0001.203E.59F0.2DE8.0000.0002  ;2167 12B1: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12B2. 0001.203E.59F0.2DE8.0000.0002  ;2168 12B2: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12B3. 0001.203E.59F0.2DE8.0000.0002  ;2169 12B3: RC[0D]_Q.Q_ID[ACC.2],RETURN2
U 12B4. 0001.203E.59F0.2DE8.0000.0002  ;2170 12B4: RC[0D]_Q.Q_ID[ACC.2],RETURN2
```

ZZ-ESKAR-V22 MODULE REVISION HISTORY
ESKARSC.MCR MICRO2 1P(00) 26-JUL-85 17:00:14


```
U 12B5. 0001.203E.59F0.2DE8.0000.0002  ;2171 12B5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B6. 0001.203E.59F0.2DE8.0000.0002  ;2172 12B6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B7. 0001.203E.59F0.2DE8.0000.0002  ;2173 12B7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B8. 0001.203E.59F0.2DE8.0000.0002  ;2174 12B8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12B9. 0001.203E.59F0.2DE8.0000.0002  ;2175 12B9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BA. 0001.203E.59F0.2DE8.0000.0002  ;2176 12BA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BB. 0001.203E.59F0.2DE8.0000.0002  ;2177 12BB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BC. 0001.203E.59F0.2DE8.0000.0002  ;2178 12BC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BD. 0001.203E.59F0.2DE8.0000.0002  ;2179 12BD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BE. 0001.203E.59F0.2DE8.0000.0002  ;2180 12BE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12BF. 0001.203E.59F0.2DE8.0000.0002  ;2181 12BF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C0. 0001.203E.59F0.2DE8.0000.0002  ;2182 12C0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C1. 0001.203E.59F0.2DE8.0000.0002  ;2183 12C1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C2. 0001.203E.59F0.2DE8.0000.0002  ;2184 12C2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C3. 0001.203E.59F0.2DE8.0000.0002  ;2185 12C3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C4. 0001.203E.59F0.2DE8.0000.0002  ;2186 12C4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C5. 0001.203E.59F0.2DE8.0000.0002  ;2187 12C5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C6. 0001.203E.59F0.2DE8.0000.0002  ;2188 12C6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C7. 0001.203E.59F0.2DE8.0000.0002  ;2189 12C7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C8. 0001.203E.59F0.2DE8.0000.0002  ;2190 12C8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12C9. 0001.203E.59F0.2DE8.0000.0002  ;2191 12C9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CA. 0001.203E.59F0.2DE8.0000.0002  ;2192 12CA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CB. 0001.203E.59F0.2DE8.0000.0002  ;2193 12CB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CC. 0001.203E.59F0.2DE8.0000.0002  ;2194 12CC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CD. 0001.203E.59F0.2DE8.0000.0002  ;2195 12CD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CE. 0001.203E.59F0.2DE8.0000.0002  ;2196 12CE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12CF. 0001.203E.59F0.2DE8.0000.0002  ;2197 12CF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D0. 0001.203E.59F0.2DE8.0000.0002  ;2198 12D0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D1. 0001.203E.59F0.2DE8.0000.0002  ;2199 12D1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D2. 0001.203E.59F0.2DE8.0000.0002  ;2200 12D2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D3. 0001.203E.59F0.2DE8.0000.0002  ;2201 12D3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D4. 0001.203E.59F0.2DE8.0000.0002  ;2202 12D4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D5. 0001.203E.59F0.2DE8.0000.0002  ;2203 12D5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D6. 0001.203E.59F0.2DE8.0000.0002  ;2204 12D6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D7. 0001.203E.59F0.2DE8.0000.0002  ;2205 12D7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D8. 0001.203E.59F0.2DE8.0000.0002  ;2206 12D8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12D9. 0001.203E.59F0.2DE8.0000.0002  ;2207 12D9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DA. 0001.203E.59F0.2DE8.0000.0002  ;2208 12DA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DB. 0001.203E.59F0.2DE8.0000.0002  ;2209 12DB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DC. 0001.203E.59F0.2DE8.0000.0002  ;2210 12DC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DD. 0001.203E.59F0.2DE8.0000.0002  ;2211 12DD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DE. 0001.203E.59F0.2DE8.0000.0002  ;2212 12DE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12DF. 0001.203E.59F0.2DE8.0000.0002  ;2213 12DF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E0. 0001.203E.59F0.2DE8.0000.0002  ;2214 12E0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E1. 0001.203E.59F0.2DE8.0000.0002  ;2215 12E1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E2. 0001.203E.59F0.2DE8.0000.0002  ;2216 12E2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E3. 0001.203E.59F0.2DE8.0000.0002  ;2217 12E3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E4. 0001.203E.59F0.2DE8.0000.0002  ;2218 12E4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E5. 0001.203E.59F0.2DE8.0000.0002  ;2219 12E5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E6. 0001.203E.59F0.2DE8.0000.0002  ;2220 12E6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E7. 0001.203E.59F0.2DE8.0000.0002  ;2221 12E7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E8. 0001.203E.59F0.2DE8.0000.0002  ;2222 12E8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12E9. 0001.203E.59F0.2DE8.0000.0002  ;2223 12E9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EA. 0001.203E.59F0.2DE8.0000.0002  ;2224 12EA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EB. 0001.203E.59F0.2DE8.0000.0002  ;2225 12EB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
```

```
U 12EC. 0001.203E.59F0.2DE8.0000.0002  ;2226 12EC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12ED. 0001.203E.59F0.2DE8.0000.0002  ;2227 12ED: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EE. 0001.203E.59F0.2DE8.0000.0002  ;2228 12EE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12EF. 0001.203E.59F0.2DE8.0000.0002  ;2229 12EF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F0. 0001.203E.59F0.2DE8.0000.0002  ;2230 12F0: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F1. 0001.203E.59F0.2DE8.0000.0002  ;2231 12F1: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F2. 0001.203E.59F0.2DE8.0000.0002  ;2232 12F2: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F3. 0001.203E.59F0.2DE8.0000.0002  ;2233 12F3: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F4. 0001.203E.59F0.2DE8.0000.0002  ;2234 12F4: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F5. 0001.203E.59F0.2DE8.0000.0002  ;2235 12F5: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F6. 0001.203E.59F0.2DE8.0000.0002  ;2236 12F6: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F7. 0001.203E.59F0.2DE8.0000.0002  ;2237 12F7: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F8. 0001.203E.59F0.2DE8.0000.0002  ;2238 12F8: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12F9. 0001.203E.59F0.2DE8.0000.0002  ;2239 12F9: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FA. 0001.203E.59F0.2DE8.0000.0002  ;2240 12FA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FB. 0001.203E.59F0.2DE8.0000.0002  ;2241 12FB: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FC. 0001.203E.59F0.2DE8.0000.0002  ;2242 12FC: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FD. 0001.203E.59F0.2DE8.0000.0002  ;2243 12FD: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FE. 0001.203E.59F0.2DE8.0000.0002  ;2244 12FE: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12FF. 0001.203E.59F0.2DE8.0000.0002  ;2245 12FF: RC[0D]_Q,Q_ID[ACC.2],RETURN2
                ;2246 ;
                ;2247 ;
                ;2248 ;*
                ;2249 ; THIS ROUTINE TRAPS TO FPA UCODE THAT  READS THE SCRATCH PAD REGISTERS
                ;2250 ; INDICATED BY THE SC.  THE CONTENTS OF RA AND RB ARE STORED IN THE FAD
                ;2251 ; DATA LATCHES. THE CONTENTS OF RB ARE BROUGHT TO BUSA BY OUTPUTING THE
                ;2252 ; FAD AR0 TO BUSA. RC[0A] IS LOADED, WITH THE REGISTER ADDRESS, FROM THE
                ;2253 ; CALLING PROGRAM.
                ;2254 ;-
                ;2255 ;
U 1169. 0818.0038.4180.0800.0000.1014  ;2256 RD.REG: D_K[.80]
U 1014. 0819.0031.0580.0800.0000.1176  ;2257 =0 D_D.OR.K[.1],CALL[GENTRA]
U 1015. 0000.003C.5980.3C00.0000.116A  ;2258  ID[ACC.2]_D
U 116A. 0000.00FC.0380.0800.0000.116B  ;2259  TRAP.FPA
U 116B. 0010.0038.0180.0950.0082.116C  ;2260  SC_RC[0A]  ; INITIALIZE SC
U 116C. 0000.003C.0180.0868.0000.116D  ;2261  LAB_R(SC)  ; FPA AT 081. AR1_RA,AR0_RB
U 116D. 0000.007C.01D8.0868.0000.116E  ;2262  LAB_R(SC),Q_ACC,CPSYNC ; READ RA
U 116E. 0001.203C.0180.09E8.0000.116F  ;2263  RC[0D]_Q  ; SAVE RA
U 116F. 0000.003C.01D8.0800.0000.1170  ;2264  Q_ACC   ; FPA AT 082. READ RB
U 1170. 0001.203E.0180.09D8.0000.0001  ;2265  RC[0B]_Q,RETURN1 ; SAVE RB
                ;2266 ;
                ;2267 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
                ;2268 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
                ;2269 ;   28: NAD/28   ; NOP
                ;2270 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
                ;2271 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
                ;2272 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
                ;2273 ;  INITIALIZE ITSELF.
                ;2274 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
                ;2275 ;
U 1171. 0818.0038.4580.0800.0000.1172  ;2276 FPINIT: D_K[.8000]
U 1172. 0000.003C.5D80.3C00.0000.101A  ;2277  ID[ACC.CS]_D  ; TURN ON FPA
U 101A. 0818.0039.2D80.0800.0000.1176  ;2278 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 101B. 0000.003C.5980.3C00.0000.1173  ;2279  ID[ACC.2]_D
U 1173. 0000.00FC.0380.0800.0000.101C  ;2280  TRAP.FPA  ; TRAP FPA TO 28.
```

```
U 101C. 0018.0039.1980.0800.0200.117C  ;2281 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 101D. 0F00.003C.0180.0800.0000.1024  ;2282  D_0
U 1024. 0000.003D.0180.0800.0000.1176  ;2283 =0 CALL,J/GENTRA
U 1025. 0000.003C.5980.3C00.0000.1174  ;2284  ID[ACC.2]_D
U 1174. 0000.00FC.0380.0800.0000.1028  ;2285  TRAP.FPA  ; TRAP FPA TO 0
U 1028. 0818.0039.2D80.0800.0000.1176  ;2286 =0 D_K[.28],CALL,J/GENTRA
U 1029. 0000.003C.5980.3C00.0000.1175  ;2287  ID[ACC.2]_D
U 1175. 0000.00FE.0380.0800.0000.0001  ;2288  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
          ;2289 ;
          ;2290 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
          ;2291 ;%0
          ;2292 ;$ 0000.0000. 0000.0000
          ;2293 ;
          ;2294 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
          ;2295 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
          ;2296 ; THE FPA FOR A TRAP TO THAT ADDRESS.
          ;2297 ;
U 1176. 0000.003C.65F8.0800.0084.7177  ;2298 GENTRA: SC_K[.10],Q_0
U 1177. 0D00.003C.8D80.0800.0084.7178  ;2299  D_DAL.SC,SC_K[.1F]
U 1178. 0801.001E.0180.0800.0000.0001  ;2300  ALU_D.ORNOT.MASK,D_ALU,RETURN1
          ;2301 ;
          ;2302 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
          ;2303 ; BY TRAPPING THE FPA TO LOCATION 0.
          ;2304 ;
          ;2305 =0
U 102A. 0F00.003D.0180.0800.0000.1176  ;2306 FPIRD: D_0,CALL,J/GENTRA
U 102B. 0000.003C.5980.3C00.0000.1179  ;2307  ID[ACC.2]_D
U 1179. 0000.00FC.0380.0800.0000.117A  ;2308  TRAP.FPA  ; TRAP TO FPA 0
U 117A. 0000.003C.0180.0800.0000.117B  ;2309  NOP
U 117B. 0000.003E.0180.0800.0000.0001  ;2310  RETURN1
          ;2311 ;
          ;2312 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
          ;2313 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
          ;2314 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
          ;2315 ;
U 117C. 2000.003C.0180.0800.0000.117D  ;2316 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 117D. 3000.003C.0180.6000.0000.117E  ;2317  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 117E. 0000.003C.0180.F800.0000.117F  ;2318 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 117F. 0000.003C.0180.F800.0000.1180  ;2319  MCT/ALLOW.IB.READ
U 1180. 0000.003C.0180.F800.0000.1181  ;2320  MCT/ALLOW.IB.READ
U 1181. 0000.003C.0180.F800.0000.1182  ;2321  MCT/ALLOW.IB.READ
U 1182. 0000.003C.0180.F800.0000.1183  ;2322  MCT/ALLOW.IB.READ
U 1183. 1000.003C.0180.F800.0000.1184  ;2323  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1184. 0000.003E.0180.0800.0000.0001  ;2324  RETURN1
U 1185. 3000.003C.0180.0800.0000.117E  ;2325 WAITIB: IBC/START,J/IBW
          ;2326 ;
          ;2327 ; THIS SUBROUTINE IS USED IN THE FPA SCRATCH PAD TEST.
          ;2328 ; IT SETS UP UNIQUE DATA PATTERNS IN EACH REGISTER SO THAT
          ;2329 ; ADDRESSING TESTS CAN BE PERFORMED. THE DATA GENERATED IS:
          ;2330 ; R[0]  00000000
          ;2331 ; R[1]  11111111
          ;2332 ; R[2]  22222222
          ;2333 ; ...
          ;2334 ; R[0E]  EEEEEEEE
          ;2335 ; R[0F]  FFFFFFFF
```

```
            ;2336 ;
U 1186. 0018.0038.61C0.0800.0000.1187  ;2337 RABTAB: Q_K[.F]
U 1187. 0000.003C.ED80.0800.0084.7188  ;2338 RABP1: SC_K[.1B]
U 1188. 0C00.003C.05F8.0800.0084.9189  ;2339  SC_SC+K[.1],D_Q,Q_0
U 1189. 0D00.003C.0180.0800.0000.118A  ;2340  D_DAL.SC
U 118A. 0F00.003C.11E0.0800.0084.718B  ;2341  Q_D,SC_K[.4],D_0
U 118B. 0D00.003C.0180.0800.0000.118C  ;2342  D_DAL.SC
U 118C. 0D00.003C.0180.0800.0000.118D  ;2343  D_DAL.SC
U 118D. 0D00.003C.0180.0800.0000.118E  ;2344  D_DAL.SC
U 118E. 0D00.003C.0180.0800.0000.118F  ;2345  D_DAL.SC
U 118F. 0D00.003C.0180.0800.0000.1190  ;2346  D_DAL.SC
U 1190. 0D00.003C.0180.0800.0000.1191  ;2347  D_DAL.SC
U 1191. 0D00.003C.0180.0800.0000.1192  ;2348  D_DAL.SC
U 1192. 0D00.003C.0180.0800.0000.1193  ;2349  D_DAL.SC
U 1193. 0019.0034.61C0.0800.0092.1194  ;2350  ALU_D[AND]K[.F],SC_ALU,Q_ALU,CLK.UBCC
U 1194. 0001.013C.0180.08E8.0000.102C  ;2351  Z?,ALU_D,R(SC)_ALU
U 102C. 0019.2000.05C0.0800.0000.1187  ;2352 =0 ALU_Q-K[.1],Q_ALU,J/RABP1
U 102D. 0000.003E.0180.0800.0000.0001  ;2353  RETURN1
            ;2354
            ;2355 ;*******************************************************
            ;2356 ;
            ;2357 ; MAIN MACHINE/FPA SUBROUTINES
            ;2358 ;
            ;2359 ;*******************************************************
            ;2360
            ;2361 ; SHIFT D REGISTTER CONTENTS
            ;2362
            ;2363
            ;2364 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
            ;2365
            ;2366
U 1195. 0000.003C.01F8.0800.0000.1196  ;2367 SHIFTD: Q_0
U 1196. 0500.003C.0180.0800.0000.1197  ;2368  D_D.LEFT  ;
U 1197. 0000.003C.0580.0800.1414.B198  ;2369  EALU_STATE-K[.1],CLK.UBCC ;
U 1198. 0000.123C.0180.0800.0000.100B  ;2370  EALU.Z?
U 100B. 0000.003C.0180.0800.0000.1195  ;2371 =1011 J/SHIFTD  ;
U 100F. 0000.003E.0180.0800.0000.0001  ;2372  RETURN1
            ;2373
            ;2374
            ;2375
            ;2376
            ;2377
            ;2378 ;*
            ;2379 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
            ;2380 ; OR'S IN BIT 15, SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP TO
            ;2381 ; THAT ADDRESS.
            ;2382 ;-
            ;2383
U 1199. 0819.0030.4580.0800.0000.119A  ;2384 TRAP1: D_D.OR.K[.8000]
U 119A. 0000.003C.65F8.0800.0084.719B  ;2385  SC_K[.10],Q_0
U 119B. 0D00.003C.0180.0800.0000.119C  ;2386  D_DAL.SC
U 119C. 0000.003C.5980.3C00.0000.119D  ;2387  ID[ACC.2]_D
U 119D. 0000.00FE.0380.0800.0000.0001  ;2388  TRAP.FPA,RETURN1
            ;2389
            ;2390
```

```
                ;2391
                ;2392
                ;2393
                ;2394
                ;2395
                ;2396 ;*
                ;2397 ; THE FOLLOWING ROUTINES GENERATE FPA UPC ADDRESSES, AND
                ;2398 ; JUMP TO THE TRAP ROUTINE.
                ;2399 ;-
                ;2400
U 119E, 0818,0038,4180,0800,0000,119F   ;2401 TRP.85: D_K[.80]
U 119F, 0819,0010,1180,0800,0000,1199   ;2402  D_D+K[.4]+1,J/TRAP1
                ;2403
                ;2404
U 11A0, 0818,0038,4180,0800,0000,11A1   ;2405 TRP.87: D_K[.80]
U 11A1, 0000,003C,5D80,0800,0000,11A2   ;2406  KMX/.7
U 11A2, 0819,0014,5D80,0800,0000,1199   ;2407  D_D+K[.7],J/TRAP1
                ;2408
                ;2409
                ;2410
                ;2411
                ;2412
                ;2413
                ;2414
U 11A3, 0818,0038,CD80,0800,0000,11A4   ;2415 TRP.F2: D_K[.F0]
U 11A4, 0819,0014,0980,0800,0000,1199   ;2416  D_D+K[2],J/TRAP1
                ;2417
                ;2418
                ;2419
                ;2420
                ;2421
U 11A5, 0818,0038,CD80,0800,0000,11A6   ;2422 TRP.F7: D_K[.F0]
U 11A6, 0000,003C,5D80,0800,0000,11A7   ;2423  KMX/.7
U 11A7, 0819,0014,5D80,0800,0000,1199   ;2424  D_D+K[.7],J/TRAP1
                ;2425
                ;2426
U 11A8, 0818,0038,CD80,0800,0000,11A9   ;2427 TRP.F8: D_K[.F0]
U 11A9, 0819,0014,0180,0800,0000,1199   ;2428  D_D+K[.8],J/TRAP1
                ;2429
                ;2430
                ;2431 ;*
                ;2432 ; THESE ROUTINES WILL LOAD THE EXPONENT AND SIGN DATA INTO THE FPA
                ;2433
                ;2434
                ;2435 ; LA<LB,SA=0,SB=0
                ;2436
U 11AA, 0818,0038,2980,0800,0000,11AB   ;2437 LDDAT2: D_K[.34]
U 11AB, 0000,003C,5DF8,0800,0084,71AC   ;2438  SC_K[.7],Q_0
U 11AC, 0D00,003C,0180,0800,0000,11AD   ;2439  D_DAL.SC
U 11AD, 0001,003C,0180,0AB0,0000,11AE   ;2440  R[6]_D
U 11AE, 0818,0038,0180,0800,0000,11AF   ;2441  D_K[.8]
U 11AF, 0000,003C,5DF8,0800,0084,71B0   ;2442  SC_K[.7],Q_0
U 11B0, 0D00,003C,0180,0800,0000,11B1   ;2443  D_DAL.SC
U 11B1, 0001,003E,0180,0AA8,0000,0001   ;2444  R[5]_D,RETURN1
                ;2445
```

```
            ;2446
            ;2447 ; LA<LB,SA=0,SB=1
            ;2448
U 11B2, 0818,0038,2D80,0800,0000,11B3  ;2449 LDDAT3: D_K[.28]
U 11B3, 0000,003C,5DF8,0800,0084,71B4  ;2450  SC_K[.7],Q_0
U 11B4, 0D00,003C,0180,0800,0000,11B5  ;2451  D_DAL.SC
U 11B5, 0001,003C,0180,0AA8,0000,11B6  ;2452  R[5]_D
U 11B6, 0818,0038,2980,0800,0000,11B7  ;2453  D_K[.34]
U 11B7, 0000,003C,5DF8,0800,0084,71B8  ;2454  SC_K[.7],Q_0
U 11B8, 0D00,003C,0180,0800,0000,11B9  ;2455  D_DAL.SC
U 11B9, 0819,0030,4580,0800,0000,11BA  ;2456  D_D.OR.K[.8000]  ; SB=1
U 11BA, 0001,003E,0180,0AB0,0000,0001  ;2457  R[6]_D,RETURN1
            ;2458
            ;2459
            ;2460 ; LA>LB,SA=1,SB=1
            ;2461
U 11BB, 0818,0038,2980,0800,0000,11BC  ;2462 LDDAT4: D_K[.34]
U 11BC, 0000,003C,5DF8,0800,0084,71BD  ;2463  SC_K[.7],Q_0
U 11BD, 0D00,003C,0180,0800,0000,11BE  ;2464  D_DAL.SC
U 11BE, 0819,0030,4580,0800,0000,11BF  ;2465  D_D.OR.K[.8000]  ; SA=1
U 11BF, 0001,003C,0180,0AA8,0000,11C0  ;2466  R[5]_D
U 11C0, 0818,0038,2D80,0800,0000,11C1  ;2467  D_K[.28]
U 11C1, 0000,003C,5DF8,0800,0084,71C2  ;2468  SC_K[.7],Q_0
U 11C2, 0D00,003C,0180,0800,0000,11C3  ;2469  D_DAL.SC
U 11C3, 0819,0030,4580,0800,0000,11C4  ;2470  D_D.OR.K[.8000]  ; SB=1
U 11C4, 0001,003E,0180,0AB0,0000,0001  ;2471  R[6]_D,RETURN1
            ;2472
            ;2473 ; LA=LB,SA=1,SB=0
            ;2474
U 11C5, 0818,0038,2980,0800,0000,11C6  ;2475 LDDAT5: D_K[.34]
U 11C6, 0000,003C,5DF8,0800,0084,71C7  ;2476  SC_K[.7],Q_0
U 11C7, 0D00,003C,0180,0800,0000,11C8  ;2477  D_DAL.SC
U 11C8, 0001,003C,0180,0AB0,0000,11C9  ;2478  R[6]_D
U 11C9, 0819,0030,4580,0800,0000,11CA  ;2479  D_D.OR.K[.8000]  ; SA=1
U 11CA, 0001,003E,0180,0AA8,0000,0001  ;2480  R[5]_D,RETURN1
            ;2481
            ;2482
            ;2483
            ;2484 ;*
            ;2485 ; THIS ROUTINE LOADS THE PR IN THE EXPONENT PROCESSOR, FROM R5.
            ;2486 ; THE PR IS ROUTED TO THE EALU OUTPUT VIA THE AMX. THE PR PROVIDES
            ;2487 ; A NON-ZERO OUTPUT WHICH PREVENTS THE NORMALIZER MUX FORCING ZEROS
            ;2488 ; ON THE BUS, WHEN THE SIGN PROCESSOR DATA IS READ BACK TO THE CPU.
            ;2489
U 11CB, 0818,0038,E980,0800,0000,11CC  ;2490 LD.PR: D_K[.24]
U 11CC, 0000,003C,0180,0800,0000,1030  ;2491  NOP
U 1030, 0819,0015,0580,0800,0000,1176  ;2492 =0 D_D*K[.1],CALL[GENTRA]
U 1031, 0000,003C,5980,3C00,0000,11CD  ;2493  ID[ACC.2]_D
U 11CD, 0000,00FC,0380,0A28,0000,11CE  ;2494  TRAP.FPA,LAB_R[5]
U 11CE, 0000,007C,0180,0A28,0000,11CF  ;2495  LAB_R[5],CPSYNC ; FPA AT 025. LOAD LA
U 11CF, 0000,003C,0180,0800,0000,11D0  ;2496  NOP  ; FPA AT 026. LOAD PR
U 11D0, 0000,003E,0180,0800,0000,0001  ;2497  RETURN1
            ;2498
            ;2499
            ;2500 ;*
```

```
            ;2501 ; THIS ROUTINE TRAPS TO FPA ADDRESS 087. AT THIS ADDRESS SA AND LA
            ;2502 ; DATA IS LOADED FROM R5. THE FPA ADDRESS IS THE ADVANCED TO 088
            ;2503 ; WHERE SB AND LB DATA IS LOADED FROM R6. R5 AND R6 ARE LOADED
            ;2504 ; IN THE CALLING PROGRAM.
            ;2505 ;-
            ;2506
            ;2507 =0
U 1032. 0000,003D,0180,0800,0000,11A0  ;2508 LD.SEX: CALL[TRP.87]
U 1033. 0000,003C,0180,0A28,0000,11D1  ;2509  LAB_R[5]  ; FPA AT 087.
U 11D1. 0000,007C,0180,0A28,0000,11D2  ;2510  LAB_R[5],CPSYNC  ; LOAD SA, LA.
U 11D2. 0000,003C,0180,0A30,0000,11D3  ;2511  LAB_R[6]  ; FPA AT 088.
U 11D3. 0000,007C,0180,0A30,0000,11D4  ;2512  LAB_R[6],CPSYNC  ; LOAD SB, LB.
U 11D4. 0000,003E,0180,0800,0000,0001  ;2513  RETURN1  ; FPA AT 028.
            ;2514
```

```
                ;2515  .PAGE "TEST 239 PRN ON RB, GENERAL REGISTERS ADDRESSING TEST 6"
                ;2516  ;******************************************************************
                ;2517  ;++
                ;2518  ; PRN ON RB, GENERAL REGISTERS ADDRESSING TEST 6
                ;2519  ;
                ;2520  ; TEST DESCRIPTION
                ;2521  ;  THIS IS TEST OF THE ADDRESSING ON THE GENERAL REGISTER
                ;2522  ;  SET COPY IN THE FPA. TESTED HERE IS THE PRN
                ;2523  ;  ADDRESS FOR THE RB REGISTERS WHICH COMES FROM THE
                ;2524  ;  INSTRUCTION BUFFER. EACH REGISTER IS FIRST LOADED
                ;2525  ;  WITH A UNIQUE DATA PATTERN:
                ;2526  ;   R[0] 00000000
                ;2527  ;   R[1] 11111111
                ;2528  ;   R[2] 22222222
                ;2529  ;    ...
                ;2530  ;   R[0F] FFFFFFFF
                ;2531  ;  THEN EACH OF THE REGISTERS 0 THRU F IS ADDRESSED
                ;2532  ;  USING THE IB PRN.
                ;2533  ;
                ;2534  ; FPA UCODE USED
                ;2535  ;  LOCATION CONTENTS
                ;2536  ;  14:  BSC/R,SCR/DP,FADC/LD,NAD/82
                ;2537  ;  82:  BSC/FAL.LH,FADC/A,NAD/82
                ;2538  ;
                ;2539  ; LOGIC DESCRIPTION
                ;2540  ;
                ;2541  ; ERROR DESCRIPTION
                ;2542  ;  EXPECTED DATA READ FROM REGISTER PRN
                ;2543  ;  GOT DATA READ FROM REGISTER PRN
                ;2544  ;  SP1 REGISTER ADDRESS
                ;2545  ;  ADDRESS IN CACHE OF OPCODE CONTAINING THAT SP1
                ;2546  ;
                ;2547  ; SYNC POINT DESCRIPTION
                ;2548  ;
                ;2549  ;--
                ;2550  ;******************************************************************
                ;2551  =0
U 1034.  0000.003D.0180.0800.0000.106E  ;2552 RABTF: NEWTST
U 1035.  0018.0038.1180.09F8.0000.1038  ;2553  RC[0F]_K[.4]
U 1038.  0018.0039.6180.09E0.0000.1186  ;2554 =0 RC[0C]_K[.F],CALL,J/RABTAB ; INITIALIZE A REGISTER
                ;2555      ; COUNTER AND PUT THE DATA INTO
                ;2556      ; THE REGISTERS.
U 1039.  0000.003C.1180.0800.0084.71D5  ;2557  SC_K[.4]  ; GENERATE ADDRESS USED TO
U 11D5.  0818.0038.65F8.0800.0000.11D6  ;2558  D_K[.10],Q_0  ; LOCATE THE OPCODES IN THE CACHE
U 11D6.  0D00.003C.0180.0800.0000.11D7  ;2559  D_DAL.SC
U 11D7.  0001.003C.0180.0980.0000.11D8  ;2560  RC[0]_D   ; RC[0]=100
U 11D8.  0019.0030.3180.09D8.0000.11D9  ;2561  RC[0B]_ALU,ALU_D.OR.K[.40]
U 11D9.  0019.0030.4180.0988.0000.11DA  ;2562  RC[1]_ALU,ALU_D.OR.K[.80]
U 11DA.  0818.0038.6580.0800.0000.103A  ;2563  D_K[.10]  ; GENERATE ADDRESS OF FPA UCODE
U 103A.  0819.0031.1180.0800.0000.1176  ;2564 =0 D_D.OR.K[.4],CALL[GENTRA]
U 103B.  0001.003C.0180.0990.0000.103C  ;2565  RC[2]_D
U 103C.  0000.003D.0180.0800.0000.115F  ;2566 =0 ERLOOP
U 103D.  0000.003C.0180.0800.0000.1040  ;2567  NOP
                ;2568 =0
U 1040.  0000.003D.0180.0800.0000.1171  ;2569 RABTF1: CALL,J/FPINIT
```

```
U 1041. 0010.0038.0180.0958.0200.1042  ;2570  VA_RC[0B]  ; LOAD THE PRN REGISTER WITH
U 1042. 0000.003D.0180.0800.0000.117C  ;2571 =0 CALL,J/LOADIB  ; TEST ADDRESS
U 1043. 0000.003C.0180.0800.0000.1004  ;2572  NOP
U 1004. 0000.003D.0180.0800.0000.1005  ;2573 =00 CALL
U 1005. 0000.003F.0180.0800.0000.1200  ;2574  SUB/SPEC,J/IRTAB
U 1006. 0000.003C.0180.0800.0000.1007  ;2575  NOP
U 1007. 0010.0038.0180.0908.0200.1044  ;2576  VA_RC[1]  ; LOAD THE OPCODE WITH THE
U 1044. 0000.003D.0180.0800.0000.117C  ;2577 =0 CALL,J/LOADIB  ; SP1=0F INTO THE IB.
U 1045. 0810.0038.0180.0910.0000.11DB  ;2578  D_RC[2]  ; TRAP THE FPA TO THE UCODE
U 11DB. 0000.003C.5980.3C00.0000.11DC  ;2579  ID[ACC.2]_D  ; WHICH REFERENCES THE REGISTER.
U 11DC. 0000.00FC.0380.0800.0000.11DD  ;2580  TRAP.FPA
U 11DD. 0000.003C.0180.0A78.0000.11DE  ;2581  LAB_R[0F]  ; TEST INSTRUCTION IN FPA.
U 11DE. 0000.003C.01D8.0800.0000.11DF  ;2582  Q_ACC  ; GET THE RESULT.
U 11DF. 0010.0038.0180.0960.0082.11E0  ;2583  SC_RC[0C]  ; GET THE EXPECTED RESULT.
U 11E0. 0800.003C.0180.0868.0000.11E1  ;2584  ALU_R(SC),D_ALU
U 11E1. 0001.003C.0180.09F0.0000.11E2  ;2585  RC[0E]_D
U 11E2. 0001.203C.0180.09E8.0000.11E3  ;2586  RC[0D]_Q
U 11E3. 001D.0000.0180.0800.0010.11E4  ;2587  ALU_D-Q,CLK.UBCC ; WAS IT CORRECT?
U 11E4. 0000.013C.0180.0800.0000.1048  ;2588  Z?
U 1048. 0000.003D.0180.0800.0000.1160  ;2589 =0 ERROR1  ; DATA FROM REGISTER WAS INCORRECT.
U 1049. 0810.0038.0180.0958.0000.11E5  ;2590  D_RC[0B]  ; INCREMENT TO OPCODE WITH NEXT SP1.
U 11E5. 0019.0014.1180.09D8.0000.11E6  ;2591  ALU_D+K[.4],RC[0B]_ALU
U 11E6. 0810.0038.0180.0960.0010.11E7  ;2592  D_RC[0C],CLK.UBCC ; TRIED ALL SPECIFIERS
U 11E7. 0000.013C.0180.0800.0000.104A  ;2593  Z?
U 104A. 0019.0000.0580.09E0.0000.1040  ;2594 =0 ALU_D-K[.1],RC[0C]_ALU,J/RABTF1 ; NO SO CONTINUE.
U 104B. 0000.003C.0180.0800.0000.104C  ;2595  NOP
          ;2596
          ;2597 ;
          ;2598 ; THIS TABLE IS USED BY ALL THE REGISTER ADDRESSING TESTS
          ;2599 ; THAT DEPEND UPON THE SPECIFIERS FROM THE IB.
          ;2600 ;
          ;2601 ;% 100
          ;2602 ;
          ;2603 ; SP2 DATA
          ;2604 ;
          ;2605 ;$ 5060.005F
          ;2606 ;$ 5060.005E
          ;2607 ;$ 5060.005D
          ;2608 ;$ 5060.005C
          ;2609 ;$ 5060.005B
          ;2610 ;$ 5060.005A
          ;2611 ;$ 5060.0059
          ;2612 ;$ 5060.0058
          ;2613 ;$ 5060.0057
          ;2614 ;$ 5060.0056
          ;2615 ;$ 5060.0055
          ;2616 ;$ 5060.0054
          ;2617 ;$ 5060.0053
          ;2618 ;$ 5060.0052
          ;2619 ;$ 5060.0051
          ;2620 ;$ 5060.0050
          ;2621 ;
          ;2622 ;% 140
          ;2623 ;
          ;2624 ; SP1 DATA
```

```
;2625 ;
;2626 ;$ 5F60.0050
;2627 ;$ 5E60.0050
;2628 ;$ 5D60.0050
;2629 ;$ 5C60.0050
;2630 ;$ 5B60.0050
;2631 ;$ 5A60.0050
;2632 ;$ 5960.0050
;2633 ;$ 5860.0050
;2634 ;$ 5760.0050
;2635 ;$ 5660.0050
;2636 ;$ 5560.0050
;2637 ;$ 5460.0050
;2638 ;$ 5360.0050
;2639 ;$ 5260.0050
;2640 ;$ 5160.0050
;2641 ;$ 5060.0050
;2642 ;
;2643 ;% 180
;2644 ;
;2645 ;$ 5F60.005F
;2646 ;
;2647
;2648
```

```
                ;2649  .PAGE "TEST 23A CPSYNC/WAIT, CLEAR UWORD TEST"
                ;2650  ;*****************************************************************
                ;2651  ;**
                ;2652  ; CPSYNC/WAIT, CLEAR UWORD TEST
                ;2653  ;
                ;2654  ; TEST DESCRIPTION
                ;2655  ;
                ;2656  ; THIS TEST VERIFIES THAT THE FPA WILL WAIT AT A GIVEN ADDRESS
                ;2657  ; IF CPSYNC IS ISSUED UNDER CODITIONS WHICH GENERATE A CLEAR
                ;2658  ; UWORD. THE TEST ALSO VERIFIES THAT THE FPA WILL CONTINUE
                ;2659  ; SEQUENCING THROUGH KNOWN ADDRESSES, WHEN THE CPSYNC IS
                ;2660  ; ISSUED UNDER NORMAL CONDITIONS.
                ;2661  ;
                ;2662  ; FPA UCODE USED
                ;2663  ; LOCATION CONTENTS
                ;2664  ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
                ;2665  ; 0DA: FADC/A,BSC/FAD.HL,NRC/LD.RR,BMXC/LB,NAD/0DB
                ;2666  ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
                ;2667  ;
                ;2668  ;;
                ;2669  ; LOGIC DESCRIPTION
                ;2670  ;
                ;2671  ; THE FPA ADDRESSING LOGIC IS ON THE FCT MODULE.
                ;2672  ;
                ;2673  ;
                ;2674  ;
                ;2675  ; ERROR DESCRIPTION
                ;2676  ;
                ;2677  ; EXPECTED NAD
                ;2678  ; RECEIVED NAD
                ;2679  ;
                ;2680  ; SYNC POINT DESCRIPTION
                ;2681  ;
                ;2682  ;--
                ;2683  ;*****************************************************************
                ;2684  =0
U 104C, 0018,0039,0980,09F8,0000,106E   ;2685 CPSTAL: NEWTST[.2]
U 104D, 0818,0038,8980,0800,0000,11E8   ;2686  D_K[.D]
U 11E8, 0000,003C,11F8,0800,0084,71E9   ;2687  SC_K[.4],Q_0
U 11E9, 0D00,003C,0180,0800,0000,11EA   ;2688  D_DAL.SC
U 11EA, 0819,0030,D980,0800,0000,11EB   ;2689  D_D.OR.K[.9]    ; ADDRESS IS 0D9.
U 11EB, 0019,0014,0580,09F0,0000,1050   ;2690  ALU_D+K[.1],RC[0E]_ALU   ; SAVE EXPECTED NEXT ADDRESS
U 1050, 0000,003D,0180,0800,0000,1176   ;2691 =0 CALL[GENTRA]
U 1051, 0000,003C,5980,3C00,0000,11EC   ;2692  ID[ACC.2]_D
U 11EC, 0000,00FC,0380,0800,0000,12AA   ;2693  TRAP.FPA
U 12AA, 0000,007C,0180,0800,0000,11ED   ;2694 12AA: CPSYNC
U 11ED, 0000,003C,0180,0800,0000,1155   ;2695  NOP
U 1155, 0000,003C,59F0,2C00,0000,11EE   ;2696 1155: Q_ID[ACC.2]
U 11EE, 0019,2034,81C0,0800,0000,11EF   ;2697  Q_Q.AND.K[.3FF]
U 11EF, 0810,0038,0180,0970,0000,11F0   ;2698  D_RC[0E]
U 11F0, 0001,203C,0180,09E8,0000,11F1   ;2699  RC[0D]_Q    ; SAVE RECEIVED NEXT ADDRESS
U 11F1, 001D,0000,0180,0800,0010,11F2   ;2700  ALU_D-Q,CLK.UBCC
U 11F2, 0000,013C,0180,0800,0000,1052   ;2701  Z?
U 1052, 0000,003D,0180,0800,0000,1160   ;2702 =0 ERROR1
                ;2703
```

```
            ;2704 ; ISSUE CPSYNC WITHOUT CPU ZERO UWORD.
            ;2705
U 1053. 0000.007C.0180.0800.0000.11F3  ;2706  CPSYNC
U 11F3. 0000.003C.59F0.2C00.0000.11F4  ;2707  Q_ID[ACC.2]
U 11F4. 0019.2034.81C0.0800.0000.11F5  ;2708  Q_Q.AND.K[.3FF]
U 11F5. 0001.203C.0180.09E8.0000.11F6  ;2709  RC[0D]_Q    ; SAVE RECEIVED NAD
U 11F6. 0818.0038.8980.0800.0000.11F7  ;2710  D_K[.D]
U 11F7. 0000.003C.11F8.0800.0084.71F8  ;2711  SC_K[.4],Q_0
U 11F8. 0D00.003C.0180.0800.0000.11F9  ;2712  D_DAL.SC
U 11F9. 0010.0038.01C0.0968.0000.11FA  ;2713  Q_RC[0D]
U 11FA. 0000.003C.F580.0800.0000.11FB  ;2714  KMX/.A
U 11FB. 0019.0010.F580.09F0.0000.11FC  ;2715  ALU_D+K[.A]+1,RC[0E]_ALU ; EXPECTED NAD IS 0DB.
U 11FC. 0810.0038.0180.0970.0000.11FD  ;2716  D_RC[0E]
U 11FD. 001D.0000.0180.0800.0010.11FE  ;2717  ALU_D-Q,CLK.UBCC
U 11FE. 0000.013C.0180.0800.0000.1054  ;2718  Z?
U 1054. 0000.003D.0180.0800.0000.1160  ;2719 =0 ERROR1
U 1055. 0000.003C.0180.0800.0000.1058  ;2720  NOP
            ;2721
```

```
;2722  .PAGE "TEST 23B LITERAL REGISTER TEST"
;2723 ;*****************************************************************
;2724 ;**
;2725 ; LITERAL REGISTER TEST
;2726 ; TEST DESCRIPTION
;2727 ;
;2728 ; THIS TEST USES A TABLE OF INSTRUCTION BUFFER DATA QUAD WORDS
;2729 ; TO BRING LONG LITERALS TO THE FPA. THE FPA IS TRAPPED TO THE
;2730 ; IRD STATE.THE INSTRUCTION BUFFER IS LOADED AND A CALL 3 IS
;2731 ; MADE. IRTAB1 CAPTURES THE CALL 3 IN THE CPU. THE FPA LEAVES
;2732 ; IRD TO PROCESS THE ADDF INSTRUCTION. THE A FORK ADDRESS FOR
;2733 ; THE FPA IS 1FA. AT 1FA THE LR IS GATED TO BUSA. IN THIS STATE
;2734 ; THE CPU GATES THE ACCELERATER DATA ON BUS A INTO THE D REGISTER.
;2735 ; A FLOATING ONE PATTERN IS USED TO TEST THE LR.
;2736 ;
;2737 ; FPA UCODE USED
;2738 ; LOCATION CONTENTS
;2739 ; 000: EALUC/3,FADC/LD,SGNC/A.RES,BSC/NH,OPLD/INIT,MSC/RR.0,
;2740 ;  NAD/0A3
;2741 ; 0A3: SCR/R.R FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,
;2742 ;  EAC/LDAB
;2743 ; 1FA: BSC/LR,FADC/AR1,EAC/LDA,SGNC/LDSA,OPLD/LDR.R,NAD/GETSP2
;2744 ;
;2745 ;
;2746 ; LOGIC DESCRIPTION
;2747 ; THE LITERAL REGISTER IS ON THE FMH AND FML MODULES. THE BUS
;2748 ; CONTROL SIGNALS ORIGINATE ON THE FCT MODULE.
;2749 ;
;2750 ;
;2751 ; ERROR DESCRIPTION
;2752 ;
;2753 ; EXPECTED LITERAL DATA
;2754 ; RECEIVED LITERAL DATA
;2755 ; CACHE ADDRESS OF LITERAL DATA
;2756 ; FPA MAINTENANCE REGISTER (BEFORE CALL 3)
;2757 ; FPA MAINTENANCE REGISTER (AFTER CALL 3)
;2758 ; LR DATA COUNT
;2759 ; CACHE ADDRESS OF DATA COUNT
;2760 ;
;2761 ; SYNC POINT DESCRIPTION
;2762 ;
;2763 ;--
;2764 ;*****************************************************************
;2765
;2766 ; NOTE THE CACHE DATA FOR THIS TEST IS AT THE END OF TEST 217.
;2767
;2768 =0
U 1058, 0018,0039,5D80,09F8,0000,106E  ;2769 LRTST: NEWTST[.7]
U 1059, 0818,0038,3180,0800,0000,11FF  ;2770  D_K[.40]
U 11FF, 0000,003C,0980,0800,0084,7300  ;2771  SC_K[.2]
U 1300, 0D00,003C,0180,0800,0000,1301  ;2772  D_DAL.SC
U 1301, 0819,0030,9D80,0800,0000,1302  ;2773  D_D.OR.K[.7C]
U 1302, 0021,003C,0180,09C0,0000,1303  ;2774  ALU_D,RC[8]_ALU.LEFT  ; DATA COUNT ADDRESS
U 1303, 0010,0038,0180,0940,0200,1304  ;2775  VA_RC[8]
U 1304, 0000,003C,0180,4000,0000,1305  ;2776  D[LONG]_CACHE
```

```
U 1305. 0001.003C.0180.09C8.0000.1306  ;2777  RC[9]_D     ; SAVE LR DATA COUNT
U 1306. 0818.0038.A580.0800.0000.1307  ;2778  D_K[.60]
U 1307. 0000.003C.0D80.0800.0084.7308  ;2779  SC_K[.3]
U 1308. 0D00.003C.0180.0800.0000.1309  ;2780  D_DAL.SC
U 1309. 0019.0014.0980.09E0.0000.105C  ;2781  ALU_D+K[.2],RC[0C]_ALU  ; ADDRESS IS 302
U 105C. 0000.003D.0180.0800.0000.115F  ;2782 =0 ERLOOP
U 105D. 0000.003C.0180.0800.0000.1060  ;2783  NOP
        ;2784 =0
U 1060. 0000.003D.0180.0800.0000.1171  ;2785 LRTST1: CALL[FPINIT]
U 1061. 0000.003C.0180.0800.0000.1062  ;2786  NOP
U 1062. 0000.003D.0180.0800.0000.102A  ;2787 =0 CALL[FPIRD]
U 1063. 0010.0038.0180.0960.0200.1064  ;2788  VA_RC[0C]    ; GET LR DATA AND OPCODE DATA
U 1064. 0000.003D.0180.0800.0000.117C  ;2789 =0 CALL[LOADIB]
U 1065. 0000.003C.0180.0800.0000.1010  ;2790  NOP
U 1010. 0000.003D.59F0.2C00.0000.1011  ;2791 =00 CALL,Q_ID[ACC.2]
U 1011. 0000.003F.0180.0800.0000.1019  ;2792  J/IRTAB1,SUB/SPEC   ; MAD=1FA
U 1012. 0001.003C.0180.09E8.0000.1013  ;2793  RC[0D]_D    ; SAVE RECEIVED LITERAL DATA
U 1013. 0001.203C.0180.09D0.0000.130A  ;2794  RC[0A]_Q    ; SAVE ACC.2 AFTER CALL 3
U 130A. 0810.0038.0180.0960.0000.130B  ;2795  D_RC[0C]
U 130B. 0019.0014.0980.09E0.0000.130C  ;2796  ALU_D+K[.2],RC[0C]_ALU  ;
U 130C. 0010.0038.0180.0960.0200.130D  ;2797  VA_RC[0C]
U 130D. 0000.003C.0180.4000.0000.130E  ;2798  D[LONG]_CACHE    ; GET EXPECTED LITERAL DATA
U 130E. 0001.003C.0180.09F0.0000.130F  ;2799  RC[0E]_D    ; SAVE EXPEXTED LR DATA
U 130F. 0010.0038.01C0.0968.0000.1310  ;2800  Q_RC[0D]    ; RECEIVED DATA
U 1310. 001D.0000.0180.0800.0010.1311  ;2801  ALU_D-Q,CLK.UBCC
U 1311. 0000.013C.0180.0800.0000.1066  ;2802  Z?
U 1066. 0000.003D.0180.0800.0000.1160  ;2803 =0 ERROR1
U 1067. 0810.0038.D580.0960.0000.1312  ;2804  D_RC[0C],KMX/.6
U 1312. 0019.0014.D580.09E0.0000.1313  ;2805  ALU_D+K[.6],RC[0C]_ALU  ; INCREMENT TO IB DATA ADDRESS
U 1313. 0810.0038.0180.0948.0010.1314  ;2806  D_RC[9],CLK.UBCC
U 1314. 0000.013C.0180.0800.0000.1068  ;2807  Z?
U 1068. 0019.0000.0580.09C8.0000.1060  ;2808 =0 ALU_D-K[.1],RC[9]_ALU,J/LRTST1
U 1069. 0000.003C.0180.0800.0000.1315  ;2809  J/ENDTST1
        ;2810
        ;2811 =00011001     ; THIS WORD IS ENTERED VIA A
U 1019. 0A01.203E.59F0.2DD8.0000.0002  ;2812 IRTAB1:RC[0B]_Q,Q_ID[ACC.2],D_ACC,RETURN2; CALL3 THEREFORE THE THE LOWER
        ;2813 =     ; BYTE IS CONSTRAINED TO 19(X)
        ;2814
U 1315. 0000.003C.0180.0800.0000.106C  ;2815 ENDTST1: NOP
        ;2816
```

```
                ;2817   .PAGE "TEST 23C SCRATCH PAD DATA TYPE TEST"
                ;2818  ;*****************************************************************
                ;2819  ;*
                ;2820  ; SCRATCH PAD DATA TYPE TEST
                ;2821  ;
                ;2822  ;
                ;2823  ; THIS TEST LOADS EACH SCRATCH PAD NIBBLE WITH THE SCRATCH PAD
                ;2824  ; ADDRESS,AND VERIFIED. EACH BYTE OF EACH SCRATCH PAD IS THEN
                ;2825  ; ASSERTED TO ZERO AND VERIFIED. ALL REGISTER NIBBLES ARE THEN
                ;2826  ; RE-LOADED AND VERIFIED AS PRVIOUSLY DESCRIBED. EACH WORD OF
                ;2827  ; EACH REGISTER IS THEN ASSERTED TO ZERO AND VERIFIED.
                ;2828  ;
                ;2829  ; FPA UCODE USED
                ;2830  ; LOCATION CONTENTS
                ;2831  ;
                ;2832  ; 081: BSC/R,FADC/LD,WAIT,FPSYNC,NAD/082
                ;2833  ; 082: BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/082
                ;2834  ;
                ;2835  ; LOGIC DESCRIPTION
                ;2836  ;
                ;2837  ; THE SCRATCH PADS ARE ON THE FNM MODULE THE ADDRESSING
                ;2838  ; LOGIC IS ON THE FCT MODULE.
                ;2839  ;
                ;2840  ;
                ;2841  ; ERROR DESCRIPTION
                ;2842  ;
                ;2843  ; EXPECTED RA (CPU)
                ;2844  ; RECEIVED RA (FPA)
                ;2845  ; EXPECTED RB (CPU)
                ;2846  ; RECEIVED RB (FPA)
                ;2847  ; REGISTER COUNTER
                ;2848  ; BYTE/WORD FLAG
                ;2849  ; BLOCK FAIL FLAG
                ;2850  ;  (1) FAILED IN RDREG1 (RB)
                ;2851  ;  (2) FAILED IN RDREG1 (RA)
                ;2852  ;  (3) FAILED IN RDREG2 (RB)
                ;2853  ;  (4) FAILED IN RDREG2 (RA)
                ;2854  ;  (6) FAILED IN RDREG3 (RB)
                ;2855  ;  (7) FAILED IN RDREG3 (RA)
                ;2856  ;
                ;2857  ; SYNC POINT DESCRIPTION
                ;2858  ;
                ;2859  ;--
                ;2860  ;*****************************************************************
                ;2861
                ;2862  =0
U 106C, 0018,0039,5D80,09F8,0000,106E   ;2863 DATYPE: NEWTST[.7]
U 106D, 0000,003C,6180,0800,0084,7316   ;2864  SC_K[.F]
U 1316, 0018,0038,1D80,09D0,0000,1317   ;2865  RC[0A]_SC  ; SAVE REGISTER COUNTER
U 1317, 0018,0038,0980,09C8,0000,1318   ;2866  RC[9]_K[.2]  ; WORD OR BYTE FLAG
                ;2867       ; 1=BYTE, 0=WORD
U 1318, 0018,0038,0180,0800,0200,1070   ;2868  VA_K[.8]
U 1070, 0000,003D,0180,0800,0000,117C   ;2869 =0 CALL[LOADIB]  ; ADDD R,R
U 1071, 0000,003C,0180,0800,0000,1072   ;2870  NOP
U 1072, 0000,003D,0180,0800,0000,115F   ;2871 =0 ERLOOP
```

```
U 1073, 0000,003C,0180,0800,0000,1074  ;2872  NOP
            ;2873
            ;2874
            ;2875 ; LOAD EACH SCRATCH PAD NIBBLE WITH THE SCRATCH PAD ADDRESS
            ;2876
            ;2877 =0
U 1074, 0000,003D,0180,0800,0000,1186  ;2878 NIB: CALL[RABTAB]
U 1075, 0000,003C,0180,0800,0000,1319  ;2879  NOP
            ;2880
            ;2881 ; NOTE: RABTAB LOADS R0 WITH ZEROS. IN ORDER TO DETECT BYTE/WORD
            ;2882 ; ASSERTED TO ZERO, R0 IS ASSERTED TO ALL ONES.
            ;2883
U 1319, 0F00,003C,0180,0800,0000,131A  ;2884  D_0
U 131A, 0001,0028,0180,0A80,0000,131B  ;2885  R[0]_NOT.D
            ;2886
            ;2887
            ;2888 ; VERIFY RA,RB.
            ;2889
U 131B, 0018,0038,6180,09D0,0000,1078  ;2890  RC[0A]_K[.F]
            ;2891 =0
U 1078, 0000,003D,0180,0800,0000,1169  ;2892 RDREG1: CALL[RD.REG]
U 1079, 0000,003C,0180,0868,0000,131C  ;2893  LAB_R(SC)
U 131C, 000C,0038,0180,09E0,0000,131D  ;2894  RC[0C]_LB  ; SAVE EXPECTED RB
U 131D, 000D,2000,0180,0800,0010,131E  ;2895  ALU_Q-LB,CLK.UBCC ; CHECK RB
U 131E, 0000,013C,0180,0800,0000,107A  ;2896  Z?
U 107A, 0018,0039,0580,09C0,0000,1160  ;2897 =0 ERROR1,RC[8]_K[.1]
U 107B, 0010,0038,01C0,0968,0000,131F  ;2898  Q_RC[0D]
U 131F, 0000,003C,0180,0868,0000,1320  ;2899  LAB_R(SC)
U 1320, 0000,003C,0180,09F0,0000,1321  ;2900  RC[0E]_LA  ; SAVE EXPECTED RA
U 1321, 001C,0000,0180,0800,0010,1322  ;2901  ALU_LA-Q,CLK.UBCC ; CHECK RA
U 1322, 0000,013C,0180,0800,0000,107C  ;2902  Z?
U 107C, 0018,0039,0980,09C0,0000,1160  ;2903 =0 ERROR1,RC[8]_K[.2]
U 107D, 0000,003C,0580,0800,0094,B323  ;2904  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 1323, 0018,0038,1D80,09D0,0000,1324  ;2905  RC[0A]_SC
U 1324, 0000,123C,0180,0800,0000,1017  ;2906  EALU.N?
U 1017, 0000,003C,0180,0800,0000,1078  ;2907 =0111 J/RDREG1
U 101F, 0810,0038,0180,0948,0000,1325  ;2908  D_RC[9]  ; DATA TYPE  BYTE OR WORD?
U 1325, 0019,0000,0580,09C8,0010,1326  ;2909  ALU_D-K[.1],RC[9]_ALU,CLK.UBCC
U 1326, 0000,013C,0180,0800,0000,1080  ;2910  Z?
U 1080, 0000,003C,0180,0800,0000,1327  ;2911 =0 J/BYTE
U 1081, 0000,003C,0180,0800,0000,1335  ;2912  J/WORD
            ;2913
            ;2914 ; LOAD EACH REGISTER BYTE WITH ZERO
            ;2915
U 1327, 0F00,003C,6180,0800,0084,7328  ;2916 BYTE: SC_K[.F],D_0  ; RE-INITIALIZE SC
U 1328, 0001,803C,0180,08E8,0000,1329  ;2917 LDREG: R(SC)_D,BYTE
U 1329, 0000,003C,0580,0800,0094,B32A  ;2918  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 132A, 0000,123C,0180,0800,0000,1027  ;2919  EALU.N?
U 1027, 0000,003C,0180,0800,0000,1328  ;2920 =0111 J/LDREG
U 102F, 0000,003C,0180,0800,0000,132B  ;2921  NOP
            ;2922
            ;2923 ; VERIFY EACH REGISTER BYTE IS ZERO
            ;2924
U 132B, 0018,0038,6180,09D0,0000,1082  ;2925  RC[0A]_K[.F]
            ;2926 =0
```

```
U 1082. 0000.003D.0180.0800.0000.1169  ;2927 RDREG2: CALL[RD.REG]
U 1083. 0000.003C.0180.0868.0000.132C  ;2928  LAB_R(SC)
U 132C. 000C.0038.0180.09E0.0000.132D  ;2929  RC[0C]_LB  ; SAVE EXPECTED RB
U 132D. 000C.8038.0180.0800.0010.132E  ;2930  ALU_LB,CLK.UBCC,BYTE ; CHECK RB
U 132E. 0000.013C.0180.0800.0000.1084  ;2931  Z?
U 1084. 0018.0039.0D80.09C0.0000.1160  ;2932 =0 ERROR1,RC[8]_K[.3]
U 1085. 0010.0038.01C0.0968.0000.132F  ;2933  Q_RC[0D]
U 132F. 0000.003C.0180.0868.0000.1330  ;2934  LAB_R(SC)
U 1330. 0000.003C.0180.09F0.0000.1331  ;2935  RC[0E]_LA   ; SAVE EXPECTED RA
U 1331. 0000.803C.0180.0800.0010.1332  ;2936  ALU_LA,CLK.UBCC,BYTE ; CHECK RA
U 1332. 0000.013C.0180.0800.0000.1086  ;2937  Z?
U 1086. 0018.0039.1180.09C0.0000.1160  ;2938 =0 ERROR1,RC[8]_K[.4]
U 1087. 0000.003C.0580.0800.0094.B333  ;2939  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 1333. 0018.0038.1D80.09D0.0000.1334  ;2940  RC[0A]_SC
U 1334. 0000.123C.0180.0800.0000.1037  ;2941  EALU.N?   ; ALL REGISTERS VERIFIED?
U 1037. 0000.003C.0180.0800.0000.1082  ;2942 =0111 J/RDREG2
U 103F. 0000.003C.0180.0800.0000.1074  ;2943  J/NIB    ; RETURN TO RABTAB
            ;2944
            ;2945 ; LOAD EACH REGISTER WORD WITH ZERO
            ;2946
U 1335. 0F00.003C.6180.0800.0084.7336  ;2947 WORD: D_0,SC_K[.F]
U 1336. 0001.403C.0180.08E8.0000.1337  ;2948 LDREG1: R(SC)_D,WORD
U 1337. 0000.003C.0580.0800.0094.B338  ;2949  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 1338. 0000.123C.0180.0800.0000.1047  ;2950  EALU.N?
U 1047. 0000.003C.0180.0800.0000.1336  ;2951 =0111 J/LDREG1
            ;2952
            ;2953 ; VERIFY EACH REGISTER WORD IS ZERO
            ;2954
U 104F. 0018.0038.6180.09D0.0000.1088  ;2955  RC[0A]_K[.F]
            ;2956 =0
U 1088. 0000.003D.0180.0800.0000.1169  ;2957 RDREG3: CALL[RD.REG]
U 1089. 0000.003C.0180.0868.0000.1339  ;2958  LAB_R(SC)
U 1339. 000C.0038.0180.09E0.0000.133A  ;2959  RC[0C]_LB  ; SAVE EXPECTED RB
U 133A. 000C.4038.0180.0800.0010.133B  ;2960  ALU_LB,CLK.UBCC,WORD ; CHECK RB
U 133B. 0000.013C.0180.0800.0000.108A  ;2961  Z?
U 108A. 0018.0039.D580.09C0.0000.1160  ;2962 =0 ERROR1,RC[8]_K[.6]
U 108B. 0010.0038.01C0.0968.0000.133C  ;2963  Q_RC[0D]
U 133C. 0000.003C.0180.0868.0000.133D  ;2964  LAB_R(SC)
U 133D. 0000.003C.0180.09F0.0000.133E  ;2965  RC[0E]_LA  ; SAVE EXPECTED RA
U 133E. 0000.403C.0180.0800.0010.133F  ;2966  ALU_LA,CLK.UBCC,WORD ; CHECK RA
U 133F. 0000.013C.0180.0800.0000.108C  ;2967  Z?
U 108C. 0018.0039.5D80.09C0.0000.1160  ;2968 =0 ERROR1,RC[8]_K[.7]
U 108D. 0000.003C.0580.0800.0094.B340  ;2969  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 1340. 0018.0038.1D80.09D0.0000.1341  ;2970  RC[0A]_SC
U 1341. 0000.123C.0180.0800.0000.1057  ;2971  EALU.N?
U 1057. 0000.003C.0180.0800.0000.1088  ;2972 =0111 J/RDREG3
U 105F. 0000.003C.0180.0800.0000.1342  ;2973  J/ENDTST
            ;2974
            ;2975
            ;2976 ; DOUBLE PRECISION IB DATA TO ALLOW READING RB VIA FAD AR0.
            ;2977
            ;2978 ;%8
            ;2979 ;$ 5060, 0040
            ;2980
U 1342. 0000.003C.0180.0800.0000.108E  ;2981 ENDTST: NOP
```

```
        :2982
        :2983
```

```
;2984  .PAGE "TEST 23D CONFLICTING  ADDRESS, SCRATCH PAD TEST"
;2985 ;*****************************************************************
;2986 ;**
;2987 ; CONFLICTING  ADDRESS,  SCRATCH PAD TEST
;2988 ;
;2989 ; TEST DESCRIPTION
;2990 ;
;2991 ; THIS TEST TRAPS TO FPA UCODE THAT READS A REGISTER SPECIFIED BY
;2992 ; THE INSTRUCTION BUFFER SP1. IN THE SAME USTATE A REGISTER
;2993 ; SPECIFIED BY THE SC, IS LOADED. THE REGISTERS ARE THEN READ BACK
;2994 ; TO THE CPU AND VERIFIED.
;2995 ;
;2996 ; FPA UCODE USED
;2997 ;
;2998 ; LOCATION CONTENTS
;2999 ; 013: BSC/R,SCR/R.R,FADC/LD,NAD/082
;3000 ; 082: BSC/FAL.LH,FADC/A,NAD/082
;3001 ; 087: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,FADC/R1,WAIT,NAD/088
;3002 ; 088: BSC/R,SCR/CPU,EAC/LB,SGNC/LDSB,FADC/BR1,WAIT,NAD/028
;3003 ; 028: NAD/028
;3004 ;
;3005 ; LOGIC DESCRIPTION
;3006 ;
;3007 ; THE SCRATCH PADS ARE ON THE FNM MODULE. THE ADDRESSING
;3008 ; LOGIC IS ON THE FCT MODULE.
;3009 ;
;3010 ;
;3011 ; ERROR DESCRIPTION
;3012 ;
;3013 ; EXPECTED REGISTER SPECIFIED BY SC
;3014 ; RECEIVED REGISTER SPECIFIED BY SC
;3015 ; EXPECTED REGISTER SPECIFIED BY IB(SP2)
;3016 ; RECEIVED REGISTER SPECIFIED BY IB(SP2)
;3017 ; REGISTER VERIFY COUNTER
;3018 ; REGISTER LOAD COUNTER
;3019 ; ADDRESS OF IB DATA
;3020 ; ADDRESS OF EXPECTED DATA
;3021 ;
;3022 ; SYNC POINT DESCRIPTION
;3023 ;
;3024 ;--
;3025 ;*****************************************************************
;3026
;3027 ; THE CACHE DATA FOR THIS TEST IS AT THE END OF TEST 217.
;3028
;3029
;3030 =0
U 108E, 0018.0039.0180.09F8.0000.106E  ;3031 CADTST: NEWTST[.8]
U 108F, 0818.0038.C980.0800.0000.1343  ;3032  D_K[.3030]
U 1343, 0000.003C.61F8.0800.0084.7344  ;3033  SC_K[.F],Q_0
U 1344, 0D00.003C.0180.0800.0000.1345  ;3034  D_DAL.SC
U 1345, 0819.0030.C980.0800.0000.1346  ;3035  D_D.OR.K[.3030]
U 1346, 0003.003C.0180.09D0.0000.1347  ;3036  RC[0A]_0  ; SAVE REGISTER VERIFY COUNTER
U 1347, 0018.0038.1D80.09C8.0000.1348  ;3037  RC[9]_SC ; TRACKS ADDRESS OF REG. LOADED WITH E7E7CFCF
U 1348, 0001.0028.0180.09A8.0000.1349  ;3038  RC[5]_NOT.D  ; CONSTANT: E7E7CFCF
```

```
U 1349. 0818.0038.3180.0800.0000.134A  ;3039  D_K[.40]
U 134A. 0001.003C.0180.09C0.0000.134B  ;3040  RC[8]_D   ; ADDRESS OF FIRST IB DATA LOAD
U 134B. 0819.0014.1180.0800.0000.134C  ;3041  ALU_D+K[.4],D_ALU ; ADDRESS OF FIRST DATA TABLE ENTRY
U 134C. 0001.003C.0180.0998.0000.134D  ;3042  RC[3]_D
U 134D. 0001.003C.0180.09B8.0000.134E  ;3043  RC[7]_D
U 134E. 0818.0038.7580.0600.0000.134F  ;3044  D_K[.20]
U 134F. 0000.003C.11F8.0800.0084.7350  ;3045  SC_K[.4],Q_0
U 1350. 0D00.003C.0180.0800.0000.1351  ;3046  D_DAL.SC
U 1351. 0001.003C.0180.09B0.0000.1090  ;3047  RC[6]_D
U 1090. 0000.003D.0180.0800.0000.115F  ;3048 =0 ERLOOP
           ;3049
           ;3050 ; LOAD EACH NIBBLE OF EACH REGISTER WITH ITS ADDRESS
           ;3051
U 1091. 0000.003C.0180.0800.0000.1092  ;3052  NOP
           ;3053 =0
U 1092. 0000.003D.0180.0800.0000.1186  ;3054 LDALL: CALL[RABTAB]
U 1093. 0000.003C.0180.0800.0000.1094  ;3055  NOP
U 1094. 0000.003D.0180.0800.0000.1171  ;3056 =0 CALL[FPINIT]
U 1095. 0010.0038.0180.0930.0200.1096  ;3057  VA_RC[6]
U 1096. 0000.003D.0180.0800.0000.117C  ;3058 =0 CALL[LOADIB]
U 1097. 0000.003C.0180.0800.0000.1020  ;3059  NOP
U 1020. 0000.003D.0180.0800.0000.1021  ;3060 =00 CALL
U 1021. 0000.003F.0180.0800.0000.1116  ;3061  SUB/SPEC,J/IRTAB2
U 1022. 0000.003C.0180.0800.0000.1023  ;3062  NOP
U 1023. 0010.0038.0180.0940.0200.1098  ;3063 REGTST: VA_RC[8]  ; GET IB DATA LOAD
U 1098. 0000.003D.0180.0800.0000.117C  ;3064 =0 CALL[LOADIB]
           ;3065
           ;3066 ; TRAP TO 013 WHERE IB SPECIFIES REGISTER ADDRESS.
           ;3067 ; NOTE: THIS READS ONE REGISTER WHILE LOADING ANOTHER.
           ;3068
U 1099. 0818.0038.2180.0800.0000.109A  ;3069  D_K[.14]
U 109A. 0819.0001.0580.0800.0000.1176  ;3070 =0 ALU_D-K[.1],D_ALU,CALL[GENTRA]
U 109B. 0010.0038.5980.3D48.0082.1352  ;3071  ID[ACC.2]_D,SC_RC[9]
U 1352. 0810.00F8.0380.0928.0000.1353  ;3072  TRAP,FPA,D_RC[5]
U 1353. 0001.003C.0180.08E8.0000.1354  ;3073  R(SC)_D   ; FPA AT 013. BUSA_R(SP1),BUSB_R(SP2),
           ;3074     ; AR_BUS, R(SC)_E7E7CFCF
U 1354. 0000.003C.01D8.0800.0000.1355  ;3075  Q_ACC   ; FPA AT 082. BUSA_AR0, R(SP2).
           ;3076
           ;3077 ; VERIFY R(SP2)
           ;3078
U 1355. 0001.203C.0180.09D8.0000.1356  ;3079  RC[0B]_Q  ; SAVE REGISTER R(SP2)
U 1356. 0010.0038.0180.0918.0200.1357  ;3080  VA_RC[3]  ; EXPECTED SP2 DATA
U 1357. 0000.003C.0180.4000.0000.1358  ;3081  D[LONG]_CACHE
U 1358. 0001.003C.0180.09E0.0000.1359  ;3082  RC[0C]_D  ; SAVE EXPECTED DATA
U 1359. 001D.2000.0180.0800.0010.135A  ;3083  ALU_Q-D,CLK.UBCC
U 135A. 0000.013C.0180.0800.0000.109C  ;3084  Z?
U 109C. 0000.003D.0180.0800.0000.1160  ;3085 =0 ERROR1
           ;3086
           ;3087 ; TRAP TO FPA 087 TO VERIFY R(SC)
           ;3088
U 109D. 0010.0038.0180.0938.0200.135B  ;3089 VFYREG: VA_RC[7]
U 135B. 0000.003C.0180.4000.0000.135C  ;3090  D[LONG]_CACHE
U 135C. 0001.003C.0180.09E0.0000.135D  ;3091  RC[0C]_D
U 135D. 0810.0038.0180.0938.0000.135E  ;3092  D_RC[7]
U 135E. 0019.0014.0180.09B8.0000.135F  ;3093  ALU_D+K[.8],RC[7]_ALU ; UPDATE DATA TABLE ADDRESS
```

```
U 135F. 0818.0038.4180.0800.0000.109E  ;3094  D_K[.80]
U 109E. 0819.0031.5D80.0800.0000.1176  ;3095 =0 D_D.OR.K[.7],CALL[GENTRA]
U 109F. 0010.0038.5980.3D50.0082.1360  ;3096  ID[ACC.2]_D,SC_RC[0A]
U 1360. 0000.00FC.0380.0800.0000.1361  ;3097  TRAP.FPA
U 1361. 0000.003C.0180.0868.0000.1362  ;3098  LAB_R(SC)  ; FPA AT 087
U 1362. 0000.003C.01D8.0868.0000.1363  ;3099  LAB_R(SC),Q_ACC  ; BUSA_R(SC)
U 1363. 0001.203C.0180.09A0.0000.1364  ;3100  RC[4]_Q  ; SAVE RECEIVED DATA
U 1364. 0818.0038.1D80.0800.0000.1365  ;3101  D_SC  ; PRESENT VALUE OF SC
U 1365. 0010.0038.01C0.0948.0000.1366  ;3102  Q_RC[9]  ; SC FOR REGISTER LOADED WITH E7E7CFCF
U 1366. 001D.2000.0180.0800.0010.1367  ;3103  ALU_Q-D,CLK.UBCC
U 1367. 0000.013C.0180.0800.0000.10A0  ;3104  Z?
U 10A0. 0000.003C.0180.0800.0000.136D  ;3105 =0 J/PAT1
            ;3106
            ;3107 ; VERIFY REGISTER LOADED WITH E7E7CFCF
            ;3108
U 10A1. 0010.0038.01C0.0920.0000.1368  ;3109  Q_RC[4]  ; RECEIVED DATA FROM R(SC)
U 1368. 0001.203C.0180.09E8.0000.1369  ;3110  RC[0D]_Q  ; SAVE
U 1369. 0810.0038.0180.0928.0000.136A  ;3111  D_RC[5]  ; EXPECTED DATA
U 136A. 0001.003C.0180.09F0.0000.136B  ;3112  RC[0E]_D  ; SAVE
U 136B. 001D.2000.0180.0800.0010.136C  ;3113  ALU_Q-D,CLK.UBCC
U 136C. 0000.013C.0180.0800.0000.10A2  ;3114  Z?
U 10A2. 0000.003D.0180.0800.0000.1160  ;3115 =0 ERROR1
U 10A3. 0000.003C.0180.0800.0000.10A5  ;3116  J/NXTREG
            ;3117
            ;3118 ; VERIFY REGISTERS LOADED BY RABTAB
            ;3119
U 136D. 0010.0038.01C0.0920.0000.136E  ;3120 PAT1: Q_RC[4]  ; RECEIVED DATA FROM R(SC)
U 136E. 0001.203C.0180.09D8.0000.136F  ;3121  RC[0B]_Q  ; SAVE
U 136F. 0810.0038.0180.0960.0000.1370  ;3122  D_RC[0C]  ; EXPECTED DATA FROM DATA TABLE
U 1370. 001D.2000.0180.0800.0010.1371  ;3123  ALU_Q-D,CLK.UBCC
U 1371. 0000.013C.0180.0800.0000.10A4  ;3124  Z?
U 10A4. 0000.003D.0180.0800.0000.1160  ;3125 =0 ERROR1
U 10A5. 0000.003C.0580.0800.0084.9372  ;3126 NXTREG: EALU_SC+K[.1],SC_EALU
U 1372. 0000.003C.6580.0800.0014.B373  ;3127  EALU_SC-K[.10],CLK.UBCC
U 1373. 0018.1238.1D80.09D0.0000.106B  ;3128  EALU.Z?,RC[0A]_SC
U 106B. 0000.003C.0180.0800.0000.109D  ;3129 =1011 J/VFYREG
U 106F. 0000.003C.0180.0800.0000.1374  ;3130  NOP
            ;3131
            ;3132
            ;3133 ; GO TO NEXT IB REGISTER SPECIFIER DATA LOAD.
            ;3134
U 1374. 0010.0038.0180.0948.0082.1375  ;3135  SC_RC[9]
U 1375. 0003.003C.0180.09D0.0000.1376  ;3136  RC[0A]_0  ; RESTORE TO FIRST R(SC)
U 1376. 0818.0038.3180.0800.0000.1377  ;3137  D_K[.40]
U 1377. 0019.0014.1180.09B8.0000.1378  ;3138  RC[7]_D+K[.4]  ; RESTORE TO FIRST DATA TABLE ENTRY
U 1378. 0810.0038.0180.0918.0000.1379  ;3139  D_RC[3]
U 1379. 0019.0014.0180.0998.0000.137A  ;3140  ALU_D+K[.8],RC[3]_ALU ; UPDATE FOR NEXT EXPECTED SP2 DATA
U 137A. 0810.0038.0180.0940.0000.137B  ;3141  D_RC[8]  ; UPDATE FOR NEXT R(SP2) READ.
U 137B. 0019.0014.0180.09C0.0000.137C  ;3142  ALU_D+K[.8],RC[8]_ALU ; UPDATE IB ADDRESS
U 137C. 0000.003C.0580.0800.0094.B37D  ;3143  EALU_SC-K[.1],SC_EALU,CLK.UBCC
U 137D. 0018.1238.1D80.09C8.0000.1077  ;3144  EALU.N?,RC[9]_SC
U 1077. 0000.003C.0180.0800.0000.1092  ;3145 =0111 J/LDALL
U 107F. 0000.003C.0180.0800.0000.137E  ;3146  J/ENTST  ;
            ;3147
            ;3148
```

```
            ;3149 =00010110    ; THIS MICRO WORD IS ENTERED VIA A CALL3
U 1116, 0A01,203E,59F0,2DD8,0000,0002  ;3150 IRTAB2: RC[0B]_Q,Q_ID[ACC.2],D_ACC,RETURN2
            ;3151      ; THEREFORE THE LOWER BYTE MUST BE 16(X)
            ;3152 =
U 137E, 0000,003C,0180,0800,0000,10A6  ;3153 ENTST: NOP
            ;3154
            ;3155
            ;3156
            ;3157
```

```
;3158  .PAGE "TEST 23E SA AND SB SIGN LATCH TEST"
;3159  ;*************************************************************************
;3160  ;**
;3161  ; TEST THE SA AND SB SIGN LATCHES
;3162  ;
;3163  ; TEST DESCRIPTION
;3164  ;
;3165  ; THE SA AND SB SIGN LATCHES ARE TESTED FOR STUCK ONE
;3166  ; AND STUCK ZERO. THE SA AND SB MUXES ARE TESTED FOR
;3167  ; SA_SB, SA_BUSA(15), AND SB_BUSB(15). THE SIGN
;3168  ; CONTROL FIELDS; SA_BUSA(15), SB_BUSB(15), SA_SA
;3169  ; AND SA_SB ARE ALSO VERIFIED.
;3170  ;
;3171  ; FPA UCODE USED
;3172  ; LOCATION CONTENTS
;3173  ; 0F2: BSC/NH,SGNC/A.B,NAD/PSTALL
;3174  ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
;3175  ; 0F7: BSC/NH,SGNC/NOP,NAD/PSTALL
;3176  ; 0F8: BSC/R,SCR/R.R,EAC/LDSB,SGNC/LDS,WAIT,AMXC/LA,NAD/0F6
;3177  ; 087: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,FADC/R1,WAIT,NAD
;3178  ; 088: BSC/R,SCR/CPU,EA/LDB,SGNC/LDSB,FADC/BR1,NAD/PSTALL
;3179  ; 085: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDSA,WAIT,NAD/0F6
;3180  ;
;3181  ; LOGIC DESCRIPTION
;3182  ; THE SIGN LATCHES,MUXES, AND SIGN FIELD DECODES ARE ALL
;3183  ; ON THE FCT MODULE. THE SIGN RESULT MUX TO BUS A IS ON
;3184  ; THE FNM MODULE.
;3185  ;
;3186  ;
;3187  ; ERROR DESCRIPTION
;3188  ; EXPECTED SIGN RESULT
;3189  ; RECEIVED SIGN RESULT
;3190  ; EXPECTED PSL N BIT
;3191  ; RECEIVED PSL N BIT
;3192  ; ERROR1 COUNTER
;3193  ;
;3194  ;
;3195  ; SYNC POINT DESCRIPTION
;3196  ;
;3197  ;--
;3198  ;*************************************************************************
;3199  ;/////////////////////////////////////////////
;3200  ;*
;3201  ; SUBTEST 1
;3202  ; TEST FOR SA=0,PSLN BIT=0 AND SA_SA
;3203  ; TEST SGNC/LDS
;3204  ;-
;3205  =0
U 10A6, 0018,0039,1180,09F8,0000,106E  ;3206 SGN1: NEWTST[.4]
U 10A7, 081D,0038,0180,0978,0000,137F  ;3207  D_RC[0F]
U 137F, 0019,0014,0580,09F8,0000,10A8  ;3208  RC[0F]_D+K[.1]
U 10A8, 000D,003D,0180,0800,0000,1164  ;3209 =0 SUBTEST
U 10A9, 000D,003C,0180,0800,0000,10AA  ;3210  NOP
U 10AA, 000D,003D,0180,0800,0000,115F  ;3211 =0 ERLOOP
U 10AB, 000D,003C,0180,0800,0000,10AC  ;3212  NOP
```

```
ZZ-ESKAR-V22  TEST 23E SA AND SB SIGN LATCH TEST                                SEQ 2422
ESKAR5C.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14          Page   8

U 10AC. 0000.003D.0180.0800.0000.11BB   ;3213 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 10AD. 0000.003C.0180.0800.0000.10AE   ;3214  NOP
U 10AE. 0000.003D.0180.0800.0000.11CB   ;3215 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10AF. 0000.003C.0180.0800.0000.10B0   ;3216  NOP
U 10B0. 0000.003D.0180.0800.0000.1032   ;3217 =0 CALL[LD.SEX]  ; LOAD SA,SB,LA,LB.
U 10B1. 0000.003C.0180.0800.0000.10B2   ;3218  NOP
U 10B2. 0000.003D.0180.0800.0000.11AA   ;3219 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 10B3. 0003.003C.0180.09F0.0000.1380   ;3220  RC[0E]_0  ; EXPECTED SA
U 1380. 0003.003C.0180.09E0.0000.1381   ;3221  RC[0C]_0  ; EXPECTED PSL N BIT
U 1381. 0018.0038.2D80.0800.0200.10B4   ;3222  VA_K[.28]
U 10B4. 0000.003D.0180.0800.0000.117C   ;3223 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 10B5. 0000.003C.0180.0800.0000.10B6   ;3224  NOP
U 10B6. 0000.003D.0180.0800.0000.11A8   ;3225 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10B7. 0000.003C.0180.0800.0000.10B8   ;3226  NOP  ; SA_BUSA(15),SB_BUSB(15),
           ;3227    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 10B8. 0000.003D.0180.0800.0000.11A5   ;3228 =0 CALL[TRP.F7]  ; SA_SA
U 10B9. 0000.003C.01D8.0800.1800.1382   ;3229  Q_ACC,MSC/LOAD.ACC.CC
U 1382. 0019.2034.45C0.0800.0000.1383   ;3230  Q_Q.AND.K[.8000]
U 1383. 0001.203C.0180.09E8.0000.1384   ;3231  RC[0D]_Q
U 1384. 0011.2020.0180.0970.0010.1385   ;3232  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1385. 0000.013C.0180.0800.0000.10BA   ;3233  Z?
U 10BA. 0018.0039.0580.09D0.0000.1160   ;3234 =0 ERROR1,RC[0A]_K[.1]
U 10BB. 0000.003C.3DF0.2C00.0000.1386   ;3235  Q_ID[PSL]
U 1386. 0019.2034.01C0.0800.0000.1387   ;3236  Q_Q.AND.K[.8]  ; MASK FOR PSL N BIT
U 1387. 0001.203C.0180.09D8.0000.1388   ;3237  RC[0B]_Q  ; RECEIVED PSL N BIT
U 1388. 0011.2020.0180.0960.0010.1389   ;3238  ALU_Q.XOR.RC[0C],CLK.UBCC
U 1389. 0000.013C.0180.0800.0000.10BC   ;3239  Z?
U 10BC. 0018.0039.0980.09D0.0000.1160   ;3240 =0 ERROR1,RC[0A]_K[.2]
U 10BD. 0000.003C.0180.0800.0000.10BE   ;3241  NOP
           ;3242 ;/////////////////////////////////////////////
           ;3243 ;+
           ;3244 ; SUBTEST 2
           ;3245 ; TEST FOR SA=1,PSL N BIT=1 AND SA_SA
           ;3246 ;-
U 10BE. 0000.003D.0180.0800.0000.1164   ;3247 =0 SUBTEST
U 10BF. 0000.003C.0180.0800.0000.10C0   ;3248  NOP
U 10C0. 0000.003D.0180.0800.0000.115F   ;3249 =0 ERLOOP
U 10C1. 0000.003C.0180.0800.0000.10C2   ;3250  NOP
U 10C2. 0000.003D.0180.0800.0000.11AA   ;3251 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 10C3. 0000.003C.0180.0800.0000.10C4   ;3252  NOP
U 10C4. 0000.003D.0180.0800.0000.11CB   ;3253 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10C5. 0000.003C.0180.0800.0000.10C6   ;3254  NOP
U 10C6. 0000.003D.0180.0800.0000.1032   ;3255 =0 CALL[LD.SEX]  ; LOAD SA,SB,LA,LB.
U 10C7. 0000.003C.0180.0800.0000.10C8   ;3256  NOP
U 10C8. 0000.003D.0180.0800.0000.11BB   ;3257 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 10C9. 0018.0038.4580.09F0.0000.138A   ;3258  RC[0E]_K[.8000]  ; EXPECTED SA
U 138A. 0018.0038.0180.09E0.0000.138B   ;3259  RC[0C]_K[.8]  ; EXPECTED PSL N BIT
U 138B. 0018.0038.2D80.0800.0200.10CA   ;3260  VA_K[.28]
U 10CA. 0000.003D.0180.0800.0000.117C   ;3261 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 10CB. 0000.003C.0180.0800.0000.10CC   ;3262  NOP
U 10CC. 0000.003D.0180.0800.0000.11A8   ;3263 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10CD. 0000.003C.0180.0800.0000.10CE   ;3264  NOP  ; SA_BUSA(15),SB_BUSB(15),
           ;3265    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 10CE. 0000.003D.0180.0800.0000.11A5   ;3266 =0 CALL[TRP.F7]  ; SA_SA
U 10CF. 0000.003C.01D8.0800.1800.138C   ;3267  Q_ACC,MSC/LOAD.ACC.CC
```

```
U 138C. 0019.2034.45C0.0800.0000.138D  ;3268  Q_Q.AND.K[.8000]
U 138D. 0001.203C.0180.09E8.0000.138E  ;3269  RC[0D]_Q
U 138E. 0011.2020.0180.0970.0010.138F  ;3270  ALU_Q.XOR.RC[0E],CLK.UBCC
U 138F. 0000.013C.0180.0800.0000.10D0  ;3271  Z?
U 10D0. 0018.0039.0580.09D0.0000.1160  ;3272 =0 ERROR1,RC[0A]_K[.1]
U 10D1. 0000.003C.3DF0.2C00.0000.1390  ;3273  Q_ID[PSL]
U 1390. 0019.2034.01C0.0800.0000.1391  ;3274  Q_Q.AND.K[.8]  ; MASK FOR PSL N BIT
U 1391. 0001.203C.0180.09D8.0000.1392  ;3275  RC[0B]_Q  ; RECEIVED PSL N BIT
U 1392. 0011.2020.0180.0960.0010.1393  ;3276  ALU_Q.XOR.RC[0C],CLK.UBCC
U 1393. 0000.013C.0180.0800.0000.10D2  ;3277  Z?
U 10D2. 0018.0039.0980.09D0.0000.1160  ;3278 =0 ERROR1,RC[0A]_K[.2]
U 10D3. 0000.003C.0180.0800.0000.10D4  ;3279  NOP
            ;3280 ;////////////////////////////////////////////
            ;3281 ;*
            ;3282 ; SUBTEST 3
            ;3283 ; TEST FOR SA=0,SA_SA
            ;3284 ; TEST SGNC/LDSA
            ;3285 ;-
U 10D4. 0000.003D.0180.0800.0000.1164  ;3286 =0 SUBTEST
U 10D5. 0000.003C.0180.0800.0000.10D6  ;3287  NOP
U 10D6. 0000.003D.0180.0800.0000.115F  ;3288 =0 ERLOOP
U 10D7. 0000.003C.0180.0800.0000.10D8  ;3289  NOP
U 10D8. 0000.003D.0180.0800.0000.11BB  ;3290 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 10D9. 0000.003C.0180.0800.0000.10DA  ;3291  NOP
U 10DA. 0000.003D.0180.0800.0000.11CB  ;3292 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10DB. 0000.003C.0180.0800.0000.10DC  ;3293  NOP
U 10DC. 0000.003D.0180.0800.0000.1032  ;3294 =0 CALL[LD.SEX]  ; LOAD SA,SAB,LA,LB.
U 10DD. 0000.003C.0180.0800.0000.10DE  ;3295  NOP
U 10DE. 0000.003D.0180.0800.0000.11AA  ;3296 =0 CALL[LDDAT2]  ; LA>LB,SA=0,SB=0
U 10DF. 0003.003C.0180.09F0.0000.1394  ;3297  RC[0E]_0  ; EXPECTED SA
U 1394. 0018.0038.2D80.0800.0200.10E0  ;3298  VA_K[.28]
U 10E0. 0000.003D.0180.0800.0000.117C  ;3299 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 10E1. 0000.003C.0180.0800.0000.10E2  ;3300  NOP
U 10E2. 0000.003D.0180.0800.0000.119E  ;3301 =0 CALL[TRP.85]  ; LOAD SA,LA,LB
U 10E3. 0000.003C.0180.0800.0000.10E4  ;3302  NOP
U 10E4. 0000.003D.0180.0800.0000.11A5  ;3303 =0 CALL[TRP.F7]  ; SA_SA
U 10E5. 0000.003C.01D8.0800.0000.1395  ;3304  Q_ACC
U 1395. 0019.2034.45C0.0800.0000.1396  ;3305  Q_Q.AND.K[.8000]
U 1396. 0001.203C.0180.09E8.0000.1397  ;3306  RC[0D]_Q
U 1397. 0011.2020.0180.0970.0010.1398  ;3307  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1398. 0000.013C.0180.0800.0000.10E6  ;3308  Z?
U 10E6. 0000.003D.0180.0800.0000.1160  ;3309 =0 ERROR1
U 10E7. 0000.003C.0180.0800.0000.10E8  ;3310  NOP
            ;3311 ;*
            ;3312 ; SUBTEST 4
            ;3313 ; TEST FOR SB=1 AND SA_SB
            ;3314 ;-
U 10E8. 0000.003D.0180.0800.0000.1164  ;3315 =0 SUBTEST
U 10E9. 0000.003C.0180.0800.0000.10EA  ;3316  NOP
U 10EA. 0000.003D.0180.0800.0000.115F  ;3317 =0 ERLOOP
U 10EB. 0000.003C.0180.0800.0000.10EC  ;3318  NOP
U 10EC. 0000.003D.0180.0800.0000.11AA  ;3319 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 10ED. 0000.003C.0180.0800.0000.10EE  ;3320  NOP
U 10EE. 0000.003D.0180.0800.0000.11CB  ;3321 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10EF. 0000.003C.0180.0800.0000.10F0  ;3322  NOP
```

```
ZZ-ESKAR-V22  TEST 23E SA AND SB SIGN LATCH TEST                                SEQ 2424
ESKARSC.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14          Page   8

U 10F0. 0000.003D.0180.0800.0000.1032  ;3323 =0 CALL[LD.SEX]  ; LOAD SA,SB,LA,LB.
U 10F1. 0000.003C.0180.0800.0000.10F2  ;3324  NOP
U 10F2. 0000.003D.0180.0800.0000.11B2  ;3325 =0 CALL[LDDAT3]  ; LA<LB,SA=0,SB=1
U 10F3. 0018.0038.4580.09F0.0000.1399  ;3326  RC[0E]_K[.8000]  ; EXPECTED SB
U 1399. 0018.0038.2D80.0800.0200.10F4  ;3327  VA_K[.28]
U 10F4. 0000.003D.0180.0800.0000.117C  ;3328 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 10F5. 0000.003C.0180.0800.0000.10F6  ;3329  NOP
U 10F6. 0000.003D.0180.0800.0000.11A8  ;3330 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10F7. 0000.003C.0180.0800.0000.10F8  ;3331  NOP  ; SA_BUSA(15),SB_BUSB(15),
          ;3332    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 10F8. 0000.003D.0180.0800.0000.11A3  ;3333 =0 CALL[TRP.F2]  ; SA_SB
U 10F9. 0000.003C.0180.0800.0000.10FA  ;3334  NOP
U 10FA. 0000.003D.0180.0800.0000.11A5  ;3335 =0 CALL[TRP.F7]  ; SA_SA
U 10FB. 0000.003C.01D8.0800.0000.139A  ;3336  Q_ACC
U 139A. 0019.2034.45C0.0800.0000.139B  ;3337  Q_Q.AND.K[.8000]
U 139B. 0001.203C.0180.09E8.0000.139C  ;3338  RC[0D]_Q
U 139C. 0011.2020.0180.0970.0010.139D  ;3339  ALU_Q.XOR.RC[0E],CLK.UBCC
U 139D. 0000.013C.0180.0800.0000.10FC  ;3340  Z?
U 10FC. 0000.003D.0180.0800.0000.1160  ;3341 =0 ERROR1
U 10FD. 0000.003C.0180.0800.0000.10FE  ;3342  NOP
          ;3343 ;/////////////////////////////////////////////
          ;3344 ;*
          ;3345 ; SUBTEST 5
          ;3346 ; TEST FOR SB=0 AND SA_SB
          ;3347 ;-
U 10FE. 0000.003D.0180.0800.0000.1164  ;3348 =0 SUBTEST
U 10FF. 0000.003C.0180.0800.0000.110A  ;3349  NOP
U 110A. 0000.003D.0180.0800.0000.115F  ;3350 =0 ERLOOP
U 110B. 0000.003C.0180.0800.0000.1110  ;3351  NOP
U 1110. 0000.003D.0180.0800.0000.11BB  ;3352 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 1111. 0000.003C.0180.0800.0000.1112  ;3353  NOP
U 1112. 0000.003D.0180.0800.0000.11CB  ;3354 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1113. 0000.003C.0180.0800.0000.1114  ;3355  NOP
U 1114. 0000.003D.0180.0800.0000.1032  ;3356 =0 CALL[LD.SEX]  ; LOAD SA,SB,LA,LB.
U 1115. 0000.003C.0180.0800.0000.1118  ;3357  NOP
U 1118. 0000.003D.0180.0800.0000.11C5  ;3358 =0 CALL[LDDAT5]  ; LA<LB,SA=1,SB=0
U 1119. 0003.003C.0180.09F0.0000.139E  ;3359  RC[0E]_0  ; EXPECTED SB
U 139E. 0018.0038.2D80.0800.0200.111A  ;3360  VA_K[.2B]
U 111A. 0000.003D.0180.0800.0000.117C  ;3361 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 111B. 0000.003C.0180.0800.0000.111C  ;3362  NOP
U 111C. 0000.003D.0180.0800.0000.11A8  ;3363 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 111D. 0000.003C.0180.0800.0000.111E  ;3364  NOP  ; SA_BUSA(15),SB_BUSB(15),
          ;3365    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 111E. 0000.003D.0180.0800.0000.11A3  ;3366 =0 CALL[TRP.F2]  ; SA_SB
U 111F. 0000.003C.0180.0800.0000.1120  ;3367  NOP
U 1120. 0000.003D.0180.0800.0000.11A5  ;3368 =0 CALL[TRP.F7]  ; SA_SA
U 1121. 0000.003C.01D8.0800.0000.139F  ;3369  Q_ACC
U 139F. 0019.2034.45C0.0800.0000.13A0  ;3370  Q_Q.AND.K[.8000]
U 13A0. 0001.203C.0180.09E8.0000.13A1  ;3371  RC[0D]_Q
U 13A1. 0011.2020.0180.0970.0010.13A2  ;3372  ALU_Q.XOR.RC[0E],CLK.UBCC
U 13A2. 0000.013C.0180.0800.0000.1122  ;3373  Z?
U 1122. 0000.003D.0180.0800.0000.1160  ;3374 =0 ERROR1
U 1123. 0000.003C.0180.0800.0000.1124  ;3375  NOP
          ;3376 ;/////////////////////////////////////////////
          ;3377 ;*
```

```
          ;3378 ; SUBTEST 6
          ;3379 ; TEST FOR SB=1 AND SA_SB
          ;3380 ; TEST FOR SGNC/LDSB
          ;3381 ;-
U 1124. 0000.003D.0180.0800.0000.1164  ;3382 =0 SUBTEST
U 1125. 0000.003C.0180.0800.0000.1126  ;3383  NOP
U 1126. 0000.003D.0180.0800.0000.115F  ;3384 =0 ERLOOP
U 1127. 0000.003C.0180.0800.0000.1128  ;3385  NOP
U 1128. 0000.003D.0180.0800.0000.11AA  ;3386 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 1129. 0000.003C.0180.0800.0000.112A  ;3387  NOP
U 112A. 0000.003D.0180.0800.0000.11CB  ;3388 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 112B. 0000.003C.0180.0800.0000.13A3  ;3389  NOP
U 13A3. 0018.0038.2D80.0800.0200.112C  ;3390  VA_K[.28]
U 112C. 0000.003D.0180.0800.0000.117C  ;3391 =0 CALL[LOADIB]  ; OPCODE IS ADDF R,R
U 112D. 0000.003C.0180.0800.0000.112E  ;3392  NOP
U 112E. 0000.003D.0180.0800.0000.11A8  ;3393 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 112F. 0000.003C.0180.0800.0000.1130  ;3394  NOP  ; SA_BUSA(15),SB_BUSB(15),
          ;3395    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 1130. 0000.003D.0180.0800.0000.11B2  ;3396 =0 CALL[LDDAT3]  ; LA<LB,SA=0,SB=1
U 1131. 0018.0038.4580.09F0.0000.1132  ;3397  RC[0E]_K[.8000]  ; EXPECTED SB
U 1132. 0000.003D.0180.0800.0000.1032  ;3398 =0 CALL[LD.SEX]  ; LOAD SA,SB,LA,LB.
U 1133. 0000.003C.0180.0800.0000.1134  ;3399  NOP
U 1134. 0000.003D.0180.0800.0000.11A3  ;3400 =0 CALL[TRP.F2]  ; SA_SB
U 1135. 0000.003C.0180.0800.0000.1136  ;3401  NOP
U 1136. 0000.003D.0180.0800.0000.11A5  ;3402 =0 CALL[TRP.F7]  ; SA_SA
U 1137. 0000.003C.01D8.0800.0000.13A4  ;3403  Q_ACC
U 13A4. 0019.2034.45C0.0800.0000.13A5  ;3404  Q_Q.AND.K[.8000]
U 13A5. 0001.203C.0180.09E8.0000.13A6  ;3405  RC[0D]_Q
U 13A6. 0011.2020.0180.0970.0010.13A7  ;3406  ALU_Q.XOR.RC[0E],CLK.UBCC
U 13A7. 0000.013C.0180.0800.0000.1138  ;3407  Z?
U 1138. 0018.0039.0580.09D0.0000.1160  ;3408 =0 ERROR1,RC[0A]_K[.1]
U 1139. 0000.003C.0180.0800.0000.13A8  ;3409  NOP
U 13A8. 0001.203C.0180.09E8.0000.13A9  ;3410  RC[0D]_Q
U 13A9. 0011.2020.0180.0970.0010.13AA  ;3411  ALU_Q.XOR.RC[0E],CLK.UBCC
U 13AA. 0000.013C.0180.0800.0000.113A  ;3412  Z?
U 113A. 0018.0039.0980.09D0.0000.1160  ;3413 =0 ERROR1,RC[0A]_K[.2]
U 113B. 0000.003C.0180.0800.0000.113C  ;3414  NOP
          ;3415
          ;3416
          ;3417
          ;3418 ;* 28
          ;3419 ;$ 5540. 0056
          ;3420
          ;3421 ; DATA FOR TEST 216.
          ;3422
          ;3423
          ;3424 ; INSTRUCTION BUFFER DATA TABLE.
          ;3425
          ;3426 ;*40
          ;3427 ;$ 5060. 0050. 0000. 0000
          ;3428 ;$ 5060. 0051. 1111. 1111
          ;3429 ;$ 5060. 0052. 2222. 2222
          ;3430 ;$ 5060. 0053. 3333. 3333
          ;3431 ;$ 5060. 0054. 4444. 4444
          ;3432 ;$ 5060. 0055. 5555. 5555
```

```
;3433 ;$ 5060, 0056, 6666, 6666
;3434 ;$ 5060, 0057, 7777, 7777
;3435 ;$ 5060, 0058, 8888, 8888
;3436 ;$ 5060, 0059, 9999, 9999
;3437 ;$ 5060, 005A, AAAA, AAAA
;3438 ;$ 5060, 005B, BBBB, BBBB
;3439 ;$ 5060, 005C, CCCC, CCCC
;3440 ;$ 5060, 005D, DDDD, DDDD
;3441 ;$ 5060, 005E, EEEE, EEEE
;3442 ;$ 5060, 005F, FFFF, FFFF
;3443
;3444
;3445 ;%200
;3446 ;$ 5F60, 005F
;3447
;3448 ; DATA FOR TEST 214.
;3449
;3450
;3451 ;% 2F8
;3452 ; LITERAL DATA COUNT
;3453 ;$ 0020,0000, 0000
;3454 ;% 300
;3455 ;$ 0000,8F40, 0000,0000
;3456 ;$ 0000,8F40, 0001,0000
;3457 ;$ 0000,8F40, 0002,0000
;3458 ;$ 0000,8F40, 0004,0000
;3459 ;$ 0000,8F40, 0008,0000
;3460 ;$ 0000,8F40, 0010,0000
;3461 ;$ 0000,8F40, 0020,0000
;3462 ;$ 0000,8F40, 0040,0000
;3463 ;$ 0000,8F40, 0080,0000
;3464 ;$ 0000,8F40, 0100,0000
;3465 ;$ 0000,8F40, 0200,0000
;3466 ;$ 0000,8F40, 0400,0000
;3467 ;$ 0000,8F40, 0800,0000
;3468 ;$ 0000,8F40, 1000,0000
;3469 ;$ 0000,8F40, 2000,0000
;3470 ;$ 0000,8F40, 4000,0000
;3471 ;$ 0000,8F40, 8000,0000
;3472 ;$ 0000,8F40, 0000,0001
;3473 ;$ 0000,8F40, 0000,0002
;3474 ;$ 0000,8F40, 0000,0004
;3475 ;$ 0000,8F40, 0000,0008
;3476 ;$ 0000,8F40, 0000,0010
;3477 ;$ 0000,8F40, 0000,0020
;3478 ;$ 0000,8F40, 0000,0040
;3479 ;$ 0000,8F40, 0000,0080
;3480 ;$ 0000,8F40, 0000,0100
;3481 ;$ 0000,8F40, 0000,0200
;3482 ;$ 0000,8F40, 0000,0400
;3483 ;$ 0000,8F40, 0000,0800
;3484 ;$ 0000,8F40, 0000,1000
;3485 ;$ 0000,8F40, 0000,2000
;3486 ;$ 0000,8F40, 0000,4000
;3487 ;$ 0000,8F40, 0000,8000
```

```
:3488
:3489
:3490
```

```
ZZ-ESKAR-V22  TEST 23F RESERVED                                                 SEQ 2428
ESKAR5C.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14           Page   8

            ;3491   .PAGE "TEST 23F RESERVED"
            ;3492 =0
U 113C, 0000,003D,0180,0800,0000,106E  ;3493 D1: NEWTST
U 113D, 0000,003C,0180,0800,0000,113E  ;3494  NOP
            ;3495
```

```
ZZ-ESKAR-V22  TEST 240 RESERVED                                                    SEQ 2429
ESKAR5C.MCR                        MICRO2  1P(00)    26-JUL-85  17:00:14          Page   8

            ;3496  .PAGE "TEST 240 RESERVED"
            ;3497 =0
U 113E. 0000.003D.0180.0800.0000.106E  ;3498 D2: NEWTST
U 113F. 0000.003C.0180.0800.0000.1140  ;3499  NOP
            ;3500
```

```
            ;3501  .PAGE "TEST 241 RESERVED"
            ;3502 =0
U 1140, 0000,003D,0180,0800,0000,106E  ;3503 D3: NEWTST
U 1141, 0000,003C,0180,0800,0000,1142  ;3504  NOP
            ;3505
```

```
ZZ-ESKAR-V22  TEST 242 RESERVED                                                    SEQ 2431
ESKARSC.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14              Page   8

            ;3506  .PAGE "TEST 242 RESERVED"
            ;3507 =0
U 1142. 0000.003D.0180.0800.0000.106E  ;3508 D4: NEWTST
U 1143. 0000.003C.0180.0800.0000.1144  ;3509  NOP
            ;3510
```

```
            ;3511  .PAGE "TEST 243 RESERVED"
            ;3512 =0
U 1144, 0000,003D,0180,0800,0000,106E  ;3513 D5: NEWTST
U 1145, 0000,003C,0180,0800,0000,1146  ;3514  NOP
            ;3515
```

```
            ;3516  .PAGE "TEST 244 RESERVED"
            ;3517 =0
U 1146. 0000.003D.0180.0800.0000.106E  ;3518 D6: NEWTST
U 1147. 0000.003C.0180.0800.0000.1148  ;3519  NOP
            ;3520
```

```
            ;3521  .PAGE "TEST 245 RESERVED"
            ;3522 =0
U 1148, 0000,003D,0180,0800,0000,106E  ;3523 D7: NEWTST
U 1149, 0000,003C,0180,0800,0000,114A  ;3524  NOP
            ;3525
```

```
            ;3526  .PAGE "TEST 246 RESERVED"
            ;3527 =0
U 114A, 0000,003D,0180,0800,0000,106E  ;3528 D8: NEWTST
U 114B, 0000,003C,0180,0800,0000,114C  ;3529  NOP
            ;3530
```

```
            ;3531  .PAGE "TEST 247 RESERVED"
            ;3532 =0
U 114C, 0000,003D,0180,0800,0000,106E  ;3533 D9: NEWTST
U 114D, 0000,003C,0180,0800,0000,114E  ;3534  NOP
            ;3535
```

```
            ;3536  .PAGE "TEST 248 RESERVED"
            ;3537 =0
U 114E, 0000,003D,0180,0800,0000,106E  ;3538 D10: NEWTST
U 114F, 0000,003C,0180,0800,0000,1150  ;3539  NOP
            ;3540 =0
U 1150, 0000,003D,0180,0800,0000,106E  ;3541 D11: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2438
ESKARSC.MCR                    MICRO2  1P(00)    26-JUL-85  17:00:14           Page   9
:                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 2573= 2574= 2575= 2576=
U 1008  1898= 1899= 1875 2371= 1918= 1919= 1876 2372=
U 1010  2791= 2792= 2793= 2794= 2257= 2258= 1877 2907=
U 1018  1878 2812= 2278= 2279= 2281= 2282= 1879 2908=
U 1020  3060= 3061= 3062= 3063= 2283= 2284= 1880 2920=
U 1028  2286= 2287= 2306= 2307= 2352= 2353= 1900 2921=
U 1030  2492= 2493= 2508= 2509= 2552= 2553= 1901 2942=
U 1038  2554= 2557= 2564= 2565= 2566= 2567= 1902 2943=
U 1040  2569= 2570= 2571= 2572= 2577= 2578= 1903 2951=
U 1048  2589= 2590= 2594= 2595= 2685= 2686= 1939: 2955=
U 1050  2691= 2692= 2702= 2706= 2719= 2720= 1904 2972=
U 1058  2769= 2770= 1942: 1905 2782= 2783= 1945: 2973=
U 1060  2785= 2786= 2787= 2788= 2789= 2790= 2803= 2804=
U 1068  2808= 2809= 1906 3129= 2863= 2864= 1916 3130=
U 1070  2869= 2870= 2871= 2872= 2878= 2879= 1917 3145=
U 1078  2892= 2893= 2897= 2898= 2903= 2904= 1920 3146=
U 1080  2911= 2912= 2927= 2928= 2932= 2933= 2938= 2939=
U 1088  2957= 2958= 2962= 2963= 2968= 2969= 3031= 3032=
U 1090  3048= 3052= 3054= 3055= 3056= 3057= 3058= 3059=
U 1098  3064= 3069= 3070= 3071= 3085= 3089= 3095= 3096=
U 10A0  3105= 3109= 3115= 3116= 3125= 3126= 3206= 3207=
U 10A8  3209= 3210= 3211= 3212= 3213= 3214= 3215= 3216=
U 10B0  3217= 3218= 3219= 3220= 3223= 3224= 3225= 3226=
U 10B8  3228= 3229= 3234= 3235= 3240= 3241= 3247= 3248=
U 10C0  3249= 3250= 3251= 3252= 3253= 3254= 3255= 3256=
U 10C8  3257= 3258= 3261= 3262= 3263= 3264= 3266= 3267=
U 10D0  3272= 3273= 3278= 3279= 3286= 3287= 3288= 3289=
U 10D8  3290= 3291= 3292= 3293= 3294= 3295= 3296= 3297=
U 10E0  3299= 3300= 3301= 3302= 3303= 3304= 3309= 3310=
U 10E8  3315= 3316= 3317= 3318= 3319= 3320= 3321= 3322=
U 10F0  3323= 3324= 3325= 3326= 3328= 3329= 3330= 3331=
U 10F8  3333= 3334= 3335= 3336= 3341= 3342= 3348= 3349=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1921 3350= 3351= 1870: 1871: 1872: 1873:
U 1110  3352= 3353= 3354= 3355= 3356= 3357= 3150= 1922
U 1118  3358= 3359= 3361= 3362= 3363= 3364= 3366= 3367=
U 1120  3368= 3369= 3374= 3375= 3382= 3383= 3384= 3385=
U 1128  3386= 3387= 3388= 3389= 3391= 3392= 3393= 3394=
U 1130  3396= 3397= 3398= 3399= 3400= 3401= 3402= 3403=
U 1138  3408= 3409= 3413= 3414= 3493= 3494= 3498= 3499=
U 1140  3503= 3504= 3508= 3509= 3513= 3514= 3518= 3519=
U 1148  3523= 3524= 3528= 3529= 3533= 3534= 3538= 3539=
U 1150  3541= 1923 1924 1925 1926 2696: 1927 1928
U 1158  1929 1930 1931 1932 1933 1934 1935 1936
U 1160  1937 1938 1940 1941 1973 1974 1980 1981
U 1168  1982 2256 2259 2260 2261 2262 2263 2264
U 1170  2265 2276 2277 2280 2285 2288 2298 2299
U 1178  2300 2308 2309 2310 2316 2317 2318 2319
U 1180  2320 2321 2322 2323 2324 2325 2337 2338
U 1188  2339 2340 2341 2342 2343 2344 2345 2346
U 1190  2347 2348 2349 2350 2351 2367 2368 2369
U 1198  2370 2384 2385 2386 2387 2388 2401 2402
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2405 2406 2407 2415 2416 2422 2423 2424
U 11A8   2427 2428 2437 2438 2439 2440 2441 2442
U 11B0   2443 2444 2449 2450 2451 2452 2453 2454
U 11B8   2455 2456 2457 2462 2463 2464 2465 2466
U 11C0   2467 2468 2469 2470 2471 2475 2476 2477
U 11C8   2478 2479 2480 2490 2491 2494 2495 2496
U 11D0   2497 2510 2511 2512 2513 2558 2559 2560
U 11D8   2561 2562 2563 2579 2580 2581 2582 2583
U 11E0   2584 2585 2586 2587 2588 2591 2592 2593
U 11E8   2687 2688 2689 2690 2693 2695 2697 2698
U 11F0   2699 2700 2701 2707 2708 2709 2710 2711
U 11F8   2712 2713 2714 2715 2716 2717 2718 2771
U 1200   1990: 1991: 1992: 1993: 1994: 1995: 1996: 1997:
U 1208   1998: 1999: 2000: 2001: 2002: 2003: 2004: 2005:
U 1210   2006: 2007: 2008: 2009: 2010: 2011: 2012: 2013:
U 1218   2014: 2015: 2016: 2017: 2018: 2019: 2020: 2021:
U 1220   2022: 2023: 2024: 2025: 2026: 2027: 2028: 2029:
U 1228   2030: 2031: 2032: 2033: 2034: 2035: 2036: 2037:
U 1230   2038: 2039: 2040: 2041: 2042: 2043: 2044: 2045:
U 1238   2046: 2047: 2048: 2049: 2050: 2051: 2052: 2053:
U 1240   2054: 2055: 2056: 2057: 2058: 2059: 2060: 2061:
U 1248   2062: 2063: 2064: 2065: 2066: 2067: 2068: 2069:
U 1250   2070: 2071: 2072: 2073: 2074: 2075: 2076: 2077:
U 1258   2078: 2079: 2080: 2081: 2082: 2083: 2084: 2085:
U 1260   2086: 2087: 2088: 2089: 2090: 2091: 2092: 2093:
U 1268   2094: 2095: 2096: 2097: 2098: 2099: 2100: 2101:
U 1270   2102: 2103: 2104: 2105: 2106: 2107: 2108: 2109:
U 1278   2110: 2111: 2112: 2113: 2114: 2115: 2116: 2117:
U 1280   2118: 2119: 2120: 2121: 2122: 2123: 2124: 2125:
U 1288   2126: 2127: 2128: 2129: 2130: 2131: 2132: 2133:
U 1290   2134: 2135: 2136: 2137: 2138: 2139: 2140: 2141:
U 1298   2142: 2143: 2144: 2145: 2146: 2147: 2148: 2149:
U 12A0   2150: 2151: 2152: 2153: 2154: 2155: 2156: 2157:
U 12A8   2158: 2159: 2694: 2161: 2162: 2163: 2164: 2165:
U 12B0   2166: 2167: 2168: 2169: 2170: 2171: 2172: 2173:
U 12B8   2174: 2175: 2176: 2177: 2178: 2179: 2180: 2181:
U 12C0   2182: 2183: 2184: 2185: 2186: 2187: 2188: 2189:
U 12C8   2190: 2191: 2192: 2193: 2194: 2195: 2196: 2197:
U 12D0   2198: 2199: 2200: 2201: 2202: 2203: 2204: 2205:
U 12D8   2206: 2207: 2208: 2209: 2210: 2211: 2212: 2213:
U 12E0   2214: 2215: 2216: 2217: 2218: 2219: 2220: 2221:
U 12E8   2222: 2223: 2224: 2225: 2226: 2227: 2228: 2229:
U 12F0   2230: 2231: 2232: 2233: 2234: 2235: 2236: 2237:
U 12F8   2238: 2239: 2240: 2241: 2242: 2243: 2244: 2245:
U 1300   2772 2773 2774 2775 2776 2777 2778 2779
U 1308   2780 2781 2795 2796 2797 2798 2799 2800
U 1310   2801 2802 2805 2806 2807 2815 2865 2866
U 1318   2868 2884 2885 2890 2894 2895 2896 2899
U 1320   2900 2901 2902 2905 2906 2909 2910 2916
U 1328   2917 2918 2919 2925 2929 2930 2931 2934
U 1330   2935 2936 2937 2940 2941 2947 2948 2949
U 1338   2950 2959 2960 2961 2964 2965 2966 2967
U 1340   2970 2971 2981 3033 3034 3035 3036 3037
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3038 3039 3040 3041 3042 3043 3044 3045
U 1350  3046 3047 3072 3073 3075 3079 3080 3081
U 1358  3082 3083 3084 3090 3091 3092 3093 3094
U 1360  3097 3098 3099 3100 3101 3102 3103 3104
U 1368  3110 3111 3112 3113 3114 3120 3121 3122
U 1370  3123 3124 3127 3128 3135 3136 3137 3138
U 1378  3139 3140 3141 3142 3143 3144 3153 3208
U 1380  3221 3222 3230 3231 3232 3233 3236 3237
U 1388  3238 3239 3259 3260 3268 3269 3270 3271
U 1390  3274 3275 3276 3277 3298 3305 3306 3307
U 1398  3308 3327 3337 3338 3339 3340 3360 3370
U 13A0  3371 3372 3373 3390 3404 3405 3406 3407
U 13A8  3410 3411 3412
U 13B0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKARSC	955
Used	955
Remaining	

Total microwords used in memory U: 955
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKARSC.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

```
ZZ-ESKAR-V22  Table of Contents                                                SEQ 2443
   ESKARSD.MCR               MICRO2  1P(00)    26-JUL-85  17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2043 TEST 249 BEN/4 BIT 2, POLY DONE, TEST
; 2094 TEST 24A BEN/4 BIT 1, CPU SYNC, TEST
; 2145 TEST 24B BEN/4 BIT 0, FLOAT OPCODE, TEST
; 2214 TEST 24C LA REGISTER, AMX/LA AND EALU/A TEST
; 2268 TEST 24D LB REGISTER, AMX/LB AND EALU/A TEST
; 2326 TEST 24E BMX/LB AND EALU/A+B, WITH A=0, TEST
; 2385 TEST 24F BMX/80 TEST
; 2435 TEST 250 XR REGISTER AND BMX/XR TEST
; 2494 TEST 251 PR REGISTER, BITS <07:00>, AND AMX/PR, BITS <07:00>, TEST
; 2556 TEST 252 EALU/K3FF TEST
; 2615 TEST 253 LA LOAD FUNCTION TEST
; 2669 TEST 254 LB LOAD FUNCTION TEST
; 2724 TEST 255 PR LOAD FUNCTION TEST
; 2782 TEST 256 XR LOAD FUNCTION TEST
; 2838 TEST 257 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 1
; 2901 TEST 258 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 2
; 2967 TEST 259 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 3
; 3043 TEST 25A PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 4
; 3128 TEST 25B PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 5
; 3218 TEST 25C BEN/7 TEST 1
; 3266 TEST 25D BEN/7 TEST 2
; 3327 TEST 25E BEN/7 TEST 3
; 3380 TEST 25F BEN/7 TEST 4
; 3441 TEST 260 BEN/7 TEST 5
; 3512 TEST 261 RESERVED
; 3517 TEST 262 RESERVED
; 3522 DUMMY NEWTST
```

```
:1  .NOLIST
:1804  .LIST
:1805


\
```

```
:1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR5D
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPEC~ED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D, 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.104F  ;1874 TRPH0: Q_ID[USTACK]
U 104F, 0C00.003C.01E0.0800.0000.105B  ;1875  Q_D,D_Q
U 105B, 0000.003C.8180.3C00.0000.105F  ;1876  ID[USTACK]_D
U 105F, 0C00.003C.01E0.0800.0000.1109  ;1877  Q_D,D_Q
U 1109, 0000.003C.01C0.0A78.0000.112B  ;1878  Q_R[0F]
U 112B, 0001.2024.0180.0800.0010.112C  ;1879  ALU_Q.ANDNOT.MASK,CLK.U9CC ; WAS TRAP EXPECTED?
U 112C, 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004. 0000.003D.D980.0800.0084.714D  ;1898 =0 SETRCF[.9]
U 1005. 0000.003C.49F0.2C00.0000.112D  ;1899  Q_ID[TBER0]
U 112D. 0001.203C.4DF0.2DB0.0000.112E  ;1900  RC[6]_Q,Q_ID[TBER1]
U 112E. 0001.203C.65F0.2DB8.0000.112F  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 112F. 0001.203C.6DF0.2DD8.0000.1130  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1130. 0001.203C.75F0.2DE0.0000.1131  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1131. 0001.203C.79F0.2DC8.0000.1132  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1132. 0001.203C.69F0.2DC0.0000.1133  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 1133. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.1147  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 1134. 0000.003C.8580.0800.0084.7135  ;1916 SCOPE: SC_K[.C]
U 1135. 0803.001C.0180.0800.0000.1006  ;1917  ALU_NOT.MASK,D_ALU
U 1006. 0000.003D.7580.3C00.0000.1150  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1007. 0000.003C.65F8.0800.0084.7136  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 1136. 0818.C038.8D80.0800.0000.1137  ;1920  D_K[.1F]    ; ...
U 1137. 0D00.003C.0180.0800.0000.1138  ;1921  D_DAL.SC
U 1138. 0000.003C.3D80.3C00.0000.1139  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 1139. 0818.0038.0580.0800.0000.113A  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 113A. 0D00.003C.0180.0800.0000.113B  ;1924  D_DAL.SC
U 113B. 0F00.003C.3180.3C00.0000.113C  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 113C. 0000.003C.0180.0800.0000.113D  ;1926  NOP
U 113D. 0000.003C.3980.3C00.0000.113E  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 113E. 0000.003C.2980.3C00.0000.113F  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 113F. 0000.003C.4980.3C00.0000.1140  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1140. 0818.0038.C180.0800.0000.1141  ;1930  D_K[.FFFF]
U 1141. 0000.003C.6580.3C00.0000.1142  ;1931  ID[SBI.ERR]_D
U 1142. 0F00.003C.6D80.3C00.0000.1143  ;1932  ID[FAULT]_D,D_0
U 1143. 0000.003C.6D80.3C00.0000.1144  ;1933  ID[FAULT]_D
U 1144. 0000.003C.6580.3C00.0000.1145  ;1934  ID[SBI.ERR]_D
U 1145. 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 1146. 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 1147. 0810.0038.0180.0978.0000.1148  ;1937 ERROR1: D_RC[0F]
U 1148. 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 1149. 0810.0038.0180.0978.0000.114A  ;1940 ERROR2: D_RC[0F]
U 114A. 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2. 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9. 0000.003C.0180.0600.0000.13FA  ;1958 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155. 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA. 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 114B. 0810.0038.4D80.0978.0000.114C  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 114C. 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 114D. 0010.0038.01C0.0978.0000.114E  ;1980 SETRCF: Q_RC[0F]
U 114E. 0019.2034.4DC0.0800.0000.114F  ;1981  Q_Q.AND.K[.FF00]
U 114F. 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983 ;
            ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;1986 ;   28: NAD/28   ; NO
            ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;1990 ;  INITIALIZE ITSELF.
            ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;1992 ;
U 1150. 0818.0038.4580.0800.0000.1151  ;1993 FPINIT: D_K[.8000]
U 1151. 0000.003C.5D80.3C00.0000.1008  ;1994  ID[ACC.CS]_D  ; TURN ON FPA
U 1008. 0818.0039.2D80.0800.0000.1156  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1009. 0000.003C.5980.3C00.0000.1152  ;1996  ID[ACC.2]_D
U 1152. 0000.00FC.0380.0800.0000.100A  ;1997  TRAP.FPA  ; TRAP FPA TO 28.
U 100A. 0018.0039.1980.0800.0200.115C  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 100B. 0F00.003C.0180.0800.0000.100C  ;1999  D_0
U 100C. 0000.003D.0180.0800.0000.1156  ;2000 =0 CALL,J/GENTRA
U 100D. 0000.003C.5980.3C00.0000.1153  ;2001  ID[ACC.2]_D
U 1153. 0000.00FC.0380.0800.0000.100E  ;2002  TRAP.FPA  ; TRAP FPA TO 0
U 100E. 0818.0039.2D80.0800.0000.1156  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 100F. 0000.003C.5980.3C00.0000.1154  ;2004  ID[ACC.2]_D
U 1154. 0000.00FE.0380.0800.0000.0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
            ;2006 ;
            ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2008 ;%0
            ;2009 ;$ 0000.0000, 0000.0000
            ;2010 ;
            ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2014 ;
U 1156. 0000.003C.65F8.0800.0084.7157  ;2015 GENTRA: SC_K[.10],Q_0
U 1157. 0D00.003C.8D80.0800.0084.7158  ;2016  D_DAL.SC.SC_K[.1F]
U 1158. 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2018 ;
            ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2021 ;
            ;2022 =0
U 1010. 0F00.003D.0180.0800.0000.1156  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 1011. 0000.003C.5980.3C00.0000.1159  ;2024  ID[ACC.2]_D
U 1159. 0000.00FC.0380.0800.0000.115A  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 115A. 0000.003C.0180.0800.0000.115B  ;2026  NOP
U 115B. 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
            ;2028 ;
            ;2029 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2030 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2031 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2032 ;
U 115C. 2000.003C.0180.0800.0000.115D  ;2033 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 115D. 3000.003C.0180.6000.0000.115E  ;2034  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 115E. 0000.003C.0180.F800.0000.115F  ;2035 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 115F. 0000.003C.0180.F800.0000.1160  ;2036  MCT/ALLOW.IB.READ
U 1160. 0000.003C.0180.F800.0000.1161  ;2037  MCT/ALLOW.IB.READ
U 1161. 0000.003C.0180.F800.0000.1162  ;2038  MCT/ALLOW.IB.READ
U 1162. 0000.003C.0180.F800.0000.1163  ;2039  MCT/ALLOW.IB.READ
U 1163. 1000.003C.0180.F800.0000.1164  ;2040  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1164. 0000.003E.0180.0800.0000.0001  ;2041  RETURN1
U 1165. 3000.003C.0180.0800.0000.115E  ;2042 WAITIB: IBC/START,J/IBW
```

```
            ;2043  .PAGE "TEST 249 BEN/4 BIT 2, POLY DONE, TEST"
            ;2044 ;*****************************************************************
            ;2045 ;*
            ;2046 ; BEN/4 BIT 2, POLY DONE, TEST"
            ;2047 ;
            ;2048 ; TEST DESCRIPTION
            ;2049 ; THIS IS A TEST OF BIT 2 OF BEN/4 WHICH IS ASSERTED
            ;2050 ; WHEN THE CPU EXECUTES A POLY DONE
            ;2051 ; DIRECTIVE USING THE ACF AND ACM FPA CONTROL FIELDS.
            ;2052 ; BEN 4 IS EXECUTED IN THE FPA BOTH WITH POLY DONE
            ;2053 ; ASSERTED AND WITH IT DEASSERTED.
            ;2054 ;
            ;2055 ; FPA UCODE USED
            ;2056 ; LOCATION CONTENTS
            ;2057 ; 15:  BEN/4,NAD/0
            ;2058 ;
            ;2059 ; LOGIC DESCRIPTION
            ;2060 ;
            ;2061 ; ERROR DESCRIPTION
            ;2062 ; EXPECTED NEXT UADDRESS
            ;2063 ; GOT NEXT UADDRESS
            ;2064 ;
            ;2065 ; SYNC POINT DESCRIPTION
            ;2066 ;
            ;2067 ;--
            ;2068 ;*****************************************************************
            ;2069 =0
U 1012, 0000.003D.0180.0800.0000.1134  ;2070 POLYD: NEWTST
U 1013, 0018.0038.0980.09F8.0000.1014  ;2071  RC[0F]_K[.2]
U 1014, 0000.003D.0180.0800.0000.1150  ;2072 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1015, 0818.0038.2180.0800.0000.1016  ;2073  D_K[.14]  ; GENERATE ADDRESS OF
U 1016, 0819.0015.0580.0800.0000.1156  ;2074 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA ; FPA UCODE USED.
U 1017, 0001.003C.5980.3E88.0000.1166  ;2075  ID[ACC.2]_D,R[1]_D
U 1166, 0818.0038.0580.0800.0000.1167  ;2076  D_K[.1]
U 1167, 0819.00D4.1380.09F0.0000.1168  ;2077  TRAP.FPA,ALU_D+K[.4],D_ALU,RC[0E]_ALU ; TRAP TO FPA LOCATION 15
U 1168, 0000.003C.59F0.2C00.0000.1169  ;2078  Q_ID[ACC.2]  ; GET RESULT OF BEN 4
U 1169, 0019.2034.81C0.09E8.0000.116A  ;2079  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 116A, 001D.0000.0180.0800.0010.116B  ;2080  ALU_D-Q,CLK.UBCC ; CHECK NAD
U 116B, 0000.013C.0180.0800.0000.1018  ;2081  Z?
U 1018, 0000.003D.0180.0800.0000.1147  ;2082 =0 ERROR1   ; NAD WAS NOT 5.
U 1019, 0000.003C.0180.0800.0000.101A  ;2083  NOP
U 101A, 0000.003D.0180.0800.0000.1150  ;2084 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 101B, 0800.003C.0180.0A08.0000.116C  ;2085  D_R[1]
U 116C, 0000.003C.5980.3C00.0000.116D  ;2086  ID[ACC.2]_D
U 116D, 0818.00F8.0780.09F0.0000.116E  ;2087  TRAP.FPA,D_K[.1],RC[0E]_K[.1] ; TRAP TO FPA 15
U 116E, 0000.00FC.5B70.2C00.0000.116F  ;2088  Q_ID[ACC.2],ACF/3,ACM/6 ; ISSUE POLY DONE AND GET NEXT UADDRESS.
U 116F, 0019.2034.81C0.09E8.0000.1170  ;2089  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 1170, 001D.0000.0180.0800.0010.1171  ;2090  ALU_D-Q,CLK.UBCC ; NAD SHOULD BE 1
U 1171, 0000.013C.0180.0800.0000.101C  ;2091  Z?
U 101C, 0000.003D.0180.0800.0000.1147  ;2092 =0 ERROR1   ; NAD WAS NOT 1.
U 101D, 0000.003C.0180.0800.0000.101E  ;2093  NOP
```

```
            ;2094  .PAGE "TEST 24A BEN/4 BIT 1, CPU SYNC, TEST"
            ;2095 ;*****************************************************************
            ;2096 ;++
            ;2097 ; BEN/4 BIT 1, CPU SYNC, TEST"
            ;2098 ;
            ;2099 ; TEST DESCRIPTION
            ;2100 ; THIS IS A TEST OF BIT 1 OF BEN/4 WHICH IS ASSERTED
            ;2101 ; WHEN THE CPU EXECUTES A CPSYNC DIRECTIVE USING
            ;2102 ; THE ACF AND ACM FPA CONTROL FIELDS. BEN 4 IS
            ;2103 ; EXECUTED IN THE FPA BOTH WITH CPSYNC ASSERTED AND
            ;2104 ; WITH IT DEASSERTED.
            ;2105 ;
            ;2106 ; FPA UCODE USED
            ;2107 ; LOCATION CONTENTS
            ;2108 ; 15:  BEN/4,NAD/0
            ;2109 ;
            ;2110 ; LOGIC DESCRIPTION
            ;2111 ;
            ;2112 ; ERROR DESCRIPTION
            ;2113 ; EXPECTED NEXT ADDRESS
            ;2114 ; GOT NEXT ADDRESS
            ;2115 ;
            ;2116 ; SYNC POINT DESCRIPTION
            ;2117 ;
            ;2118 ;--
            ;2119 ;*****************************************************************
            ;2120 =0
U 101E, 0000.003D.0180.0800.0000.1134  ;2121 BCPSYN: NEWTST
U 101F, 0018.0038.0980.09F8.0000.1020  ;2122  RC[0F]_K[.2]
U 1020, 0000.003D.0180.0800.0000.1150  ;2123 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 1021, 0818.0038.2180.0800.0000.1022  ;2124  D_K[.14]  ; GENERATE ADDRESS OF FPA UCODE.
U 1022, 0819.0015.0580.0800.0000.1156  ;2125 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 1023, 0001.003C.5980.3E88.0000.1172  ;2126  ID[ACC.2]_D,R[1]_D
U 1172, 0818.0038.0580.0800.0000.1173  ;2127  D_K[.1]
U 1173, 0819.00D4.1380.09F0.0000.1174  ;2128  TRAP.FPA,ALU_D+K[.4],D_ALU,RC[0E]_ALU ; TRAP TO LOCATION 15
U 1174, 0000.003C.59F0.2C00.0000.1175  ;2129  Q_ID[ACC.2]  ; GET RESULT OF BEN/4 IN NAD FIELD.
U 1175, 0019.2034.81C0.09E8.0000.1176  ;2130  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 1176, 001D.0000.0180.0800.0010.1177  ;2131  ALU_D-Q,CLK.UBCC
U 1177, 0000.013C.0180.0800.0000.1024  ;2132  Z?
U 1024, 0000.003D.0180.0800.0000.1147  ;2133 =0 ERROR1   ; NAD WAS NOT 5.
U 1025, 0000.003C.0180.0800.0000.1026  ;2134  NOP
U 1026, 0000.003D.0180.0800.0000.1150  ;2135 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1027, 0800.003C.0180.0A08.0000.1178  ;2136  D_R[1]
U 1178, 0000.003C.5980.3C00.0000.1179  ;2137  ID[ACC.2]_D
U 1179, 0818.00F8.5F80.09F0.0000.117A  ;2138  TRAP.FPA,D_K[.7],RC[0E]_K[.7] ; TRAP TO FPA 15
U 117A, 0000.007C.59F0.2C00.0000.117B  ;2139  Q_ID[ACC.2],CPSYNC ; ISSUE CPSYNC AND GET NAD
U 117B, 0019.2034.81C0.09E8.0000.117C  ;2140  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 117C, 001D.0000.0180.0800.0010.117D  ;2141  ALU_D-Q,CLK.UBCC
U 117D, 0000.013C.0180.0800.0000.1028  ;2142  Z?
U 1028, 0000.003D.0180.0800.0000.1147  ;2143 =0 ERROR1   ; NAD WAS NOT 7
U 1029, 0000.003C.0180.0800.0000.102A  ;2144  NOP
```

```
                ;2145  .PAGE "TEST 24B BEN/4 BIT 0, FLOAT OPCODE, TEST"
                ;2146 ;*****************************************************************
                ;2147 ;**
                ;2148 ; BEN/4 BIT 0, FLOAT OPCODE, TEST
                ;2149 ;
                ;2150 ; TEST DESCRIPTION
                ;2151 ; THIS IS A TEST OF BIT 0 OF BEN/4 WHICH IS ASSERTED
                ;2152 ; WHEN THE OPCODE IN THE IB HAS BIT 5 CLEAR. THIS BIT
                ;2153 ; (OPCODE<5>) DISTINGUISHES A FLOATING OPCODE FROM
                ;2154 ; A DOUBLE OPCODE WHICH HAS BIT 5 SET. BEN/4 IS EXECUTED
                ;2155 ; WITH TWO DIFFERENT OPCODES:
                ;2156 ;  FFFFFFDF
                ;2157 ;  00000020
                ;2158 ; AND THE RESULT OF THE BRANCH IS CHECKED FOR EACH.
                ;2159 ;
                ;2160 ; FPA UCODE USED
                ;2161 ; LOCATION CONTENTS
                ;2162 ; 15:  BEN/4,NAD/0
                ;2163 ;
                ;2164 ; LOGIC DESCRIPTION
                ;2165 ;
                ;2166 ; ERROR DESCRIPTION
                ;2167 ; EXPECTED NEXT ADDRESS
                ;2168 ; GOT NEXT ADDRESS
                ;2169 ; OPCODE IN IB
                ;2170 ;
                ;2171 ; SYNC POINT DESCRIPTION
                ;2172 ;
                ;2173 ;--
                ;2174 ;*****************************************************************
                ;2175 =0
U 102A, 0000.003D.0180.0800.0000.1134  ;2176 FLOP: NEWTST
U 102B, 0018.0038.0D80.09F8.0084.717E  ;2177  RC[0F]_K[.3],SC_K[.3]
U 117E, 0000.003C.0980.0800.0084.917F  ;2178  SC_SC+K[.2]
U 117F, 0000.0038.0180.09E0.0000.102C  ;2179  ALU_MASK,RC[0C]_ALU ; SAVE OPCODE=FFFFFFDF
U 102C, 0000.003D.0180.0800.0000.1150  ;2180 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 102D, 0018.0038.7580.0800.0200.102E  ;2181  VA_K[.20]  ; SET ADDRESS TO REFERENCE OPCODE.
U 102E, 0000.003D.0180.0800.0000.115C  ;2182 =0 CALL,J/LOADIB  ; LOAD OPCODE INTO IB
U 102F, 0818.0038.2180.0800.0000.1030  ;2183  D_K[.14]  ; GENERATE ADDRESS OF FPA UCODE.
U 1030, 0819.0015.0580.0800.0000.1156  ;2184 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 1031, 0001.003C.5980.3E88.0000.1180  ;2185  ID[ACC.2]_D,R[1]_D
U 1180, 0818.0038.0580.0800.0000.1181  ;2186  D_K[.1]
U 1181, 0819.00D4.1380.09F0.0000.1182  ;2187  TRAP.FPA,ALU_D+K[.4],D_ALU,RC[0E]_ALU  ; TRAP TO FPA 15
U 1182, 0000.003C.59F0.2C00.0000.1183  ;2188  Q_ID[ACC.2]  ; GET RESULT OF BEN/4
U 1183, 0019.2034.81C0.09E8.0000.1184  ;2189  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 1184, 001D.0000.0180.0800.0010.1185  ;2190  ALU_D-Q,CLK.UBCC
U 1185, 0000.013C.0180.0800.0000.1032  ;2191  Z?
U 1032, 0000.003D.0180.0800.0000.1147  ;2192 =0 ERROR1   ; NEXT ADDRESS NOT 5
U 1033, 0810.0038.0180.0960.0000.1186  ;2193  D_RC[0C]
U 1186, 0001.0028.0180.09E0.0000.1034  ;2194  ALU_NOT.D,RC[0C]_ALU ; OPCODE=00000020
U 1034, 0000.003D.0180.0800.0000.1150  ;2195 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 1035, 0018.0038.7580.0800.0200.1036  ;2196  VA_K[.20]  ; GENERATE ADDRESS OF OPCODE.
U 1036, 0000.003D.0180.0803.0000.115C  ;2197 =0 VA_VA+4,CALL,J/LOADIB ; LOAD THE OPCODE INTO THE IB.
U 1037, 0800.003C.0180.0A08.0000.1187  ;2198  D_R[1]
U 1187, 0000.003C.5980.3C00.0000.1188  ;2199  ID[ACC.2]_D
```

```
U 1188, 0018,00F8,1380,09F0,0000,1189  ;2200  TRAP.FPA,RC[0E]_K[.4] ; TRAP TO FPA 15
U 1189, 0000,003C,59F0,2C00,0000,118A  ;2201  Q_ID[ACC.2]  ; GET THE RESULT OF THE BEN/4
U 118A, 0019,2034,81C0,09E8,0000,118B  ;2202  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 118B, 0019,2000,1180,0800,0010,118C  ;2203  ALU_Q-K[.4],CLK.UBCC ; CORRECT?
U 118C, 0000,013C,0180,0800,0000,1038  ;2204  Z?
U 1038, 0000,003D,0180,0800,0000,1147  ;2205 =0 ERROR1   ; NEXT ADDRESS NOT 4
U 1039, 0000,003C,0180,0800,0000,103A  ;2206  NOP
          ;2207 ;
          ;2208 ; THESE OPCODES ARE IN THE CACHE TO BE USED BY THE BEN/4 BIT 0 TEST:
          ;2209 ;
          ;2210 ;x20
          ;2211 ;$ FFDF,FFFF
          ;2212 ;$ 0020,0000
          ;2213 ;
```

```
            ;2214  .PAGE "TEST 24C LA REGISTER, AMX/LA AND EALU/A TEST"
            ;2215 ;*****************************************************************
            ;2216 ;**
            ;2217 ; LA REGISTER, AMX/LA AND EALU/A TEST
            ;2218 ;
            ;2219 ; TEST DESCRIPTION
            ;2220 ; THIS TEST FLOATS A 1 ACROSS A FIELD OF 0'S THRU THE
            ;2221 ; LA REGISTER AND THE PATH THRU THE AMX AND EALU.
            ;2222 ;
            ;2223 ; FPA UCODE USED
            ;2224 ; LOCATION CONTENTS
            ;2225 ; 16:  BSC/R,SCR/CPU,EAC/LDA,NAD/17,WAIT
            ;2226 ; 17:  AMXC/LA,EALUC/A,BSC/NH,NAD/PSTALL
            ;2227 ;
            ;2228 ; LOGIC DESCRIPTION
            ;2229 ;
            ;2230 ; ERROR DESCRIPTION
            ;2231 ; EXPECTED RESULT (IN BITS <14:07>)
            ;2232 ; GOT RESULT (IN BITS <14:07>)
            ;2233 ;
            ;2234 ; SYNC POINT DESCRIPTION
            ;2235 ;
            ;2236 ;--
            ;2237 ;*****************************************************************
            ;2238 =0
U 103A, 0000.003D.0180.0800.0000.1134  ;2239 LATST: NEWTST
U 103B, 0018.0038.0980.09F8.0000.118D  ;2240  RC[0F]_K[.2]
U 118D, 0000.003C.5D80.0800.0084.718E  ;2241  SC_K[.7]  ; GENERATE A MASK USED
U 118E, 0803.0010.6180.0800.0084.718F  ;2242  ALU_MASK+1,D_ALU,SC_K[.F] ; TO CALCULATE THE
U 118F, 0003.0010.01C0.0800.0000.1190  ;2243  ALU_MASK+1,Q_ALU ; EXPECTED RESULT.
U 1190, 001D.0024.0180.0A88.0000.1191  ;2244  ALU_D[ANDNOT]Q,R[1]_ALU ; R[1]=7F80
U 1191, 0018.0038.8D80.0A98.0000.103C  ;2245  R[3]_K[.1F]  ; SET UP A COUNTER
U 103C, 0000.003D.0180.0800.0000.1146  ;2246 =0 ERLOOP
U 103D, 0000.003C.0180.0800.0000.103E  ;2247  NOP
            ;2248 =0
U 103E, 0000.003D.0180.0800.0000.1150  ;2249 LATST1: CALL,J/FPINIT
U 103F, 0818.0038.2180.0800.0000.1040  ;2250  D_K[.14]  ; GENERATE ADDRESS OF
U 1040, 0819.0015.0980.0800.0000.1156  ;2251 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 1041, 0000.003C.5980.3E18.0082.1192  ;2252  ID[ACC.2]_D,SC_R[3] ; GENERATE TEST OPERAND.
U 1192, 0803.001C.0180.0A80.0000.1193  ;2253  ALU_NOT.MASK,R[0]_ALU,D_ALU
U 1193, 080D.0034.0180.0A08.0000.1194  ;2254  ALU_D[AND]R[1],D_ALU ; CALCULATE EXPECTED RESULT.
U 1194, 0001.00FC.0380.09F0.0000.1195  ;2255  TRAP,FPA,RC[0E]_D ; TRAP TO FPA 16
U 1195, 0000.003C.0180.0A00.0000.1196  ;2256  LAB_R[0]  ; LA LOADED IN FPA HERE.
U 1196, 0000.007C.0180.0A00.0000.1197  ;2257  LAB_R[0],CPSYNC
U 1197, 0000.003C.01D8.0800.0000.1198  ;2258  Q_ACC  ; LA READ IN FPA HERE
U 1198, 000D.2034.01C0.0A08.0000.1199  ;2259  ALU_Q[AND]R[1],Q_ALU
U 1199, 0001.203C.0180.09E8.0000.119A  ;2260  RC[0D]_Q
U 119A, 001D.0000.0180.0800.0010.119B  ;2261  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 119B, 0000.013C.0180.0800.0000.1042  ;2262  Z?
U 1042, 0000.003D.0180.0800.0000.1147  ;2263 =0 ERROR1  ; INCORRECT!
U 1043, 0000.003C.0180.0A18.0010.119C  ;2264  ALU_R[3],CLK.UBCC ; FINISHED TEST?
U 119C, 0000.013C.0180.0800.0000.1044  ;2265  Z?
U 1044, 0018.0000.0580.0A98.0000.103E  ;2266 =0 R[3]_LA-K[.1],J/LATST1 ; HAVEN'T TRIED ALL OPERANDS, CONTINUE.
U 1045, 0000.003C.0180.0800.0000.1046  ;2267  NOP
```

```
                ;2268  .PAGE "TEST 24D LB REGISTER, AMX/LB AND EALU/A TEST"
                ;2269  ;*****************************************************************
                ;2270  ;**
                ;2271  ; LB REGISTER, AMX/LB AND EALU/A TEST
                ;2272  ;
                ;2273  ; TEST DESCRIPTION
                ;2274  ; THIS TEST FLOATS A 1 ACROSS 0'S IN THE LB REGISTER
                ;2275  ; AND THE LB INPUTS TO THE AMX.
                ;2276  ;
                ;2277  ; FPA UCODE USED
                ;2278  ; LOCATION CONTENTS
                ;2279  ; 18:  BSC/R,SCR/CPU,EAC/LDB,NAD/19,WAIT
                ;2280  ; 19:  BSC/R,SCR/CPU,EAC/LDA,NAD/1A,WAIT
                ;2281  ; 1A:  AMXC/LB,EALUC/A,BSC/NH,NAD/PSTALL
                ;2282  ;
                ;2283  ; LOGIC DESCRIPTION
                ;2284  ;
                ;2285  ; ERROR DESCRIPTION
                ;2286  ; EXPECTED DATA (IN BITS <14:07>)
                ;2287  ; GOT DATA (IN BITS <14:07>)
                ;2288  ;
                ;2289  ; SYNC POINT DESCRIPTION
                ;2290  ;
                ;2291  ;--
                ;2292  ;*****************************************************************
                ;2293  =0
U 1046, 0000,003D,0180,0800,0000,1134  ;2294 LBTST: NEWTST
U 1047, 0018,0038,0980,09F8,0000,119D  ;2295  RC[0F]_K[.2]
U 119D, 0000,003C,5D80,0800,0084,719E  ;2296  SC_K[.7]  ; GENERATE A MASK USED
U 119E, 0803,0010,6180,0800,0084,719F  ;2297  ALU_MASK*1,D_ALU,SC_K[.F] ; TO CALCULATE THE
U 119F, 0003,0010,01C0,0800,0000,11A0  ;2298  ALU_MASK*1,Q_ALU ; EXPECTED RESULT.
U 11A0, 001D,0024,0180,0A88,0000,11A1  ;2299  ALU_D[ANDNOT]Q,R[1]_ALU ; R[1]=7F80
U 11A1, 0018,0038,8D80,0A98,0000,1048  ;2300  R[3]_K[.1F]  ; SET UP A COUNTER
U 1048, 0000,003D,0180,0800,0000,1146  ;2301 =0 ERLOOP
U 1049, 0000,003C,0180,0800,0000,104A  ;2302  NOP
                ;2303 =0
U 104A, 0000,003D,0180,0800,0000,1150  ;2304 LBTST1: CALL,J/FPINIT
U 104B, 0818,0038,2180,0800,0000,104C  ;2305  D_K[.14]  ; GENERATE ADDRESS OF
U 104C, 0819,0015,1180,0800,0000,1156  ;2306 =0 ALU_D*K[.4],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 104D, 0000,003C,5980,3E18,0082,11A2  ;2307  ID[ACC.2]_D,SC_R[3] ; GENERATE TEST OPERAND.
U 11A2, 0803,001C,0180,0A80,0000,11A3  ;2308  ALU_NOT.MASK,R[0]_ALU,D_ALU
U 11A3, 080D,0034,0180,0A08,0000,11A4  ;2309  ALU_D[AND]R[1],D_ALU ; CALCULATE EXPECTED RESULT.
U 11A4, 0003,003C,0180,0AA8,0000,11A5  ;2310  R[5]_0
U 11A5, 0001,00FC,0380,09F0,0000,11A6  ;2311  TRAP,FPA,RC[0E]_D ; TRAP TO FPA 18
U 11A6, 0000,003C,0180,0A00,0000,11A7  ;2312  LAB_R[0]  ; LB LOADED IN FPA HERE.
U 11A7, 0000,007C,0180,0A00,0000,11A8  ;2313  LAB_R[0],CPSYNC
U 11A8, 0000,003C,0180,0A28,0000,11A9  ;2314  LAB_R[5]  ; LA LOADED WITH 0 IN FPA HERE.
U 11A9, 0000,007C,0180,0A28,0000,11AA  ;2315  LAB_R[5],CPSYNC
U 11AA, 0000,003C,01D8,0800,0000,11AB  ;2316  Q_ACC  ; LB READ IN FPA HERE
U 11AB, 000D,2034,01C0,0A08,0000,11AC  ;2317  ALU_Q[AND]R[1],Q_ALU
U 11AC, 0001,203C,0180,09E8,0000,11AD  ;2318  RC[0D]_Q
U 11AD, 001D,0000,0180,0800,0010,11AE  ;2319  ALU_D-Q,CLK,UBCC ; RESULT CORRECT?
U 11AE, 0000,013C,0180,0800,0000,1050  ;2320  Z?
U 1050, 0000,003D,0180,0800,0000,1147  ;2321 =0 ERROR1  ; INCORRECT!
U 1051, 0000,003C,0180,0A18,0010,11AF  ;2322  ALU_R[3],CLK,UBCC ; FINISHED TEST?
```

```
U 11AF, 0000.013C.0180.0800.0000.1052  ;2323  Z?
U 1052, 0018.0000.0580.0A98.0000.104A  ;2324 =0 R[3]_LA-K[.1],J/LBTST1 ; HAVEN'T TRIED ALL OPERANDS, CONTINUE.
U 1053, 0000.003C.0180.0800.0000.1054  ;2325  NOP
```

```
            ;2326  .PAGE "TEST 24E BMX/LB AND EALU/A+B, WITH A=0, TEST"
            ;2327 ;*****************************************************************
            ;2328 ;++
            ;2329 ; BMX/LB AND EALU/A+B, WITH A=0, TEST
            ;2330 ;
            ;2331 ; TEST DESCRIPTION
            ;2332 ; THIS IS A TEST OF THE LB INPUTS TO THE BMX AND
            ;2333 ; THE BMX INPUTS TO THE EALU. NOTE THAT NO EALU/B FUNCTION
            ;2334 ; EXISTS SO THAT THE RESULT IS OBSERVED BY DOING
            ;2335 ; AN EALU/A+B FUNCTION WITH THE A INPUT=0.
            ;2336 ;
            ;2337 ; FPA UCODE USED
            ;2338 ; LOCATION CONTENTS
            ;2339 ; 1C:  BSC/R,SCR/CPU,EAC/LDB,NAD/1D,WAIT
            ;2340 ; 1D:  BSC/R,SCR/CPU,EAC/LDA,NAD/1E,WAIT
            ;2341 ; 1E:  AMXC/LA,BMXC/LB,EALUC/A+B,BSC/NH,NAD/1E
            ;2342 ;
            ;2343 ; LOGIC DESCRIPTION
            ;2344 ;
            ;2345 ; ERROR DESCRIPTION
            ;2346 ; EXPECTED RESULT (BITS<14:07>)
            ;2347 ; GOT RESULT (BITS<14:07>)
            ;2348 ;
            ;2349 ; SYNC POINT DESCRIPTION
            ;2350 ;
            ;2351 ;--
            ;2352 ;*****************************************************************
            ;2353 =0
U 1054,  0000.003D.0180.0800.0000.1134  ;2354 LBMX: NEWTST
U 1055,  0018.0038.0980.09F8.0000.11B0  ;2355  RC[0F]_K[.2]
U 11B0,  0000.003C.5D80.0800.0084.71B1  ;2356  SC_K[.7]  ; GENERATE A MASK USED
U 11B1,  0803.0010.6180.0800.0084.71B2  ;2357  ALU_MASK+1,D_ALU,SC_K[.F] ; TO CALCULATE THE
U 11B2,  0003.0010.01C0.0800.0000.11B3  ;2358  ALU_MASK+1,Q_ALU ; EXPECTED RESULT.
U 11B3,  001D.0024.0180.0A88.0000.11B4  ;2359  ALU_D[ANDNOT]Q,R[1]_ALU ; R[1]=7F80
U 11B4,  0018.0038.8D80.0A98.0000.1056  ;2360  R[3]_K[.1F]  ; SET UP A COUNTER
U 1056,  0000.003D.0180.0800.0000.1146  ;2361 =0 ERLOOP
U 1057,  0003.003C.0180.0AA8.0000.1058  ;2362  R[5]_0
            ;2363 =0
U 1058,  0000.003D.0180.0800.0000.1150  ;2364 LBMX1: CALL,J/FPINIT
U 1059,  0818.0038.E580.0800.0000.105C  ;2365  D_K[.1A]  ; GENERATE ADDRESS OF
U 105C,  0819.0015.0980.0800.0000.1156  ;2366 =0 ALU_D+K[.2],D_ALU,CALL,J/CENTRA ; FPA UCODE.
U 105D,  0000.003C.5980.3E18.0082.11B5  ;2367  ID[ACC.2]_D,SC_R[3] ; GENERATE TEST OPERAND.
U 11B5,  0803.001C.0180.0A80.0000.11B6  ;2368  ALU_NOT.MASK,R[0]_ALU,D_ALU
U 11B6,  080D.0034.0180.0A08.0000.11B7  ;2369  ALU_D[AND]R[1],D_ALU ; CALCULATE EXPECTED RESULT.
U 11B7,  0001.00FC.0380.09F0.0000.11B8  ;2370  TRAP,FPA,RC[0E]_D ; TRAP TO FPA 1C
U 11B8,  0000.003C.0180.0A00.0000.11B9  ;2371  LAB_R[0]  ; LB LOADED IN FPA HERE.
U 11B9,  0000.007C.0180.0A00.0000.11BA  ;2372  LAB_R[0],CPSYNC
U 11BA,  0000.003C.0180.0A28.0000.11BB  ;2373  LAB_R[5]  ; LA LOADED WITH 0 HERE
U 11BB,  0000.007C.0180.0A28.0000.11BC  ;2374  LAB_R[5],CPSYNC
U 11BC,  0000.003C.01D8.0800.0000.11BD  ;2375  Q_ACC  ; LB READ IN FPA HERE
U 11BD,  000D.2034.01C0.0A08.0000.11BE  ;2376  ALU_Q[AND]R[1],Q_ALU
U 11BE,  0001.203C.0180.09E8.0000.11BF  ;2377  RC[0D]_Q
U 11BF,  001D.0000.0180.0800.0010.11C0  ;2378  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 11C0,  0000.013C.0180.0800.0000.1060  ;2379  Z?
U 1060,  0000.003D.0180.0800.0000.1147  ;2380 =0 ERROR1  ; INCORRECT!
```

```
U 1061, 0000,003C,0180,0A18,0010,11C1  ;2381  ALU_R[3],CLK.UBCC ; FINISHED TEST?
U 11C1, 0000,013C,0180,0800,0000,1062  ;2382  Z?
U 1062, 0018,0000,0580,0A98,0000,1058  ;2383 =0 R[3]_LA-K[.1],J/LBMX1 ; HAVEN'T TRIED ALL OPERANDS, CONTINUE.
U 1063, 0000,003C,0180,0800,0000,1064  ;2384  NOP
```

```
            ;2385  .PAGE "TEST 24F BMX/80 TEST"
            ;2386 ;*****************************************************************
            ;2387 ;**
            ;2388 ; BMX/80 TEST
            ;2389 ;
            ;2390 ; TEST DESCRIPTION
            ;2391 ; THIS IS A TEST OF THE BMX FUNCTION WHICH PUTS THE CONSTANT
            ;2392 ; 80 ON THE BMX OUTPUT.
            ;2393 ;
            ;2394 ; FPA UCODE USED
            ;2395 ; LOCATION CONTENTS
            ;2396 ; 1F:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6B,WAIT
            ;2397 ; 6B:  BSC/R,SCR/CPU,EAC/LDA,NAD/20,WAIT
            ;2398 ; 20:  BMXC/#80,AMXC/LA,EALUC/A+B,BSC/NH,NAD/20
            ;2399 ;
            ;2400 ; LOGIC DESCRIPTION
            ;2401 ;
            ;2402 ; ERROR DESCRIPTION
            ;2403 ; EXPECTED RESULT (IN BITS<14:07>)
            ;2404 ; GOT RESULT (IN BITS<14:07>)
            ;2405 ;
            ;2406 ; SYNC POINT DESCRIPTION
            ;2407 ;
            ;2408 ;--
            ;2409 ;*****************************************************************
            ;2410 =0
U 1064, 0000,003D,0180,0800,0000,1134  ;2411 BMX80: NEWTST
U 1065, 0018,0038,0980,09F8,0084,71C2  ;2412  RC[0F]_K[.2],SC_K[.2] ; GENERATE THE EXPECTED RESULT.
U 11C2, 0000,003C,8580,0800,0084,91C3  ;2413  SC_SC+K[.C]
U 11C3, 0000,0038,0180,0A80,0000,1066  ;2414  ALU_MASK,R[0]_ALU
U 1066, 0003,001D,0180,09F0,0000,1150  ;2415 =0 ALU_NOT.MASK,RC[0E]_ALU,CALL,J/FPINIT ; INITIALIZE THE FPA.
U 1067, 0000,003C,5D80,0800,0084,71C4  ;2416  SC_K[.7]  ; GENERATE A MASK USED TO TEST THE
U 11C4, 0803,0010,6180,0800,0084,71C5  ;2417  ALU_MASK+1,D_ALU,SC_K[.F] ; RESULT.
U 11C5, 0003,0010,01C0,0800,0000,11C6  ;2418  ALU_MASK+1,Q_ALU
U 11C6, 001D,0024,0180,0A88,0000,1068  ;2419  ALU_D[ANDNOT]Q,R[1]_ALU ; R[1]=00007F80
U 1068, 0818,0039,8D80,0800,0000,1156  ;2420 =0 D_K[.1F],CALL,J/GENTRA ; GENERATE ADDRESS OF FPA UCODE.
U 1069, 0000,003C,5980,3C00,0000,11C7  ;2421  ID[ACC.2]_D
U 11C7, 0003,00FC,0380,0AA8,0000,11C8  ;2422  TRAP,FPA,R[5]_0  ; TRAP TO FPA LOCATION 1F.
U 11C8, 0000,003C,0180,0A00,0000,11C9  ;2423  LAB_R[0]  ; LOAD 00003F80 INTO LB
U 11C9, 0000,007C,0180,0A00,0000,11CA  ;2424  LAB_R[0],CPSYNC
U 11CA, 0000,003C,0180,0A28,0000,11CB  ;2425  LAB_R[5]  ; LOAD 0 INTO LA
U 11CB, 0000,007C,0180,0A28,0000,11CC  ;2426  LAB_R[5],CPSYNC
U 11CC, 0000,003C,01D8,0800,0000,11CD  ;2427  Q_ACC  ; GET RESULT
U 11CD, 0800,003C,0180,0A08,0000,11CE  ;2428  D_R[1]
U 11CE, 001D,0034,01C0,09E8,0000,11CF  ;2429  ALU_D[AND]Q,Q_ALU,RC[0D]_ALU
U 11CF, 0810,0038,0180,0970,0000,11D0  ;2430  D_RC[0E]
U 11D0, 001D,0000,0180,0800,0010,11D1  ;2431  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 11D1, 0000,013C,0180,0800,0000,106A  ;2432  Z?
U 106A, 0000,003D,0180,0800,0000,1147  ;2433 =0 ERROR1  ; BMXC/#80 FAILED.
U 106B, 0000,003C,0180,0800,0000,106C  ;2434  NOP
```

```
            ;2435  .PAGE "TEST 250 XR REGISTER AND BMX/XR TEST"
            ;2436 ;*****************************************************************
            ;2437 ;**
            ;2438 ; XR REGISTER AND BMX/XR TEST
            ;2439 ;
            ;2440 ; TEST DESCRIPTION
            ;2441 ; THIS IS A TEST OF THE XR REGISTER AND ITS INPUTS TO
            ;2442 ; THE BMX. A 1 IS FLOATED ACROSS 0'S TO GENERATE THE
            ;2443 ; TEST DATA.
            ;2444 ;
            ;2445 ; FPA UCODE USED
            ;2446 ; LOCATION CONTENTS
            ;2447 ; 21:  BSC/R,SCR/CPU,EAC/LDA,NAD/22,WAIT
            ;2448 ; 22:  AMXC/LA,EALUC/A,EAC/PR.XR,NAD/23
            ;2449 ; 23:  BSC/R,SCR/CPU,EAC/LDAB,NAD/24,WAIT
            ;2450 ; 24:  BMXC/XR,AMXC/LA,EALUC/A+B,BSC/NH,NAD/24
            ;2451 ;
            ;2452 ; LOGIC DESCRIPTION
            ;2453 ;
            ;2454 ; ERROR DESCRIPTION
            ;2455 ; EXPECTED RESULT (BITS<14:07>)
            ;2456 ; GOT RESULT (BITS<14:07>)
            ;2457 ;
            ;2458 ; SYNC POINT DESCRIPTION
            ;2459 ;
            ;2460 ;--
            ;2461 ;*****************************************************************
            ;2462 =0
U 106C, 0000.003D.0180.0800.0000.1134  ;2463 XRT: NEWTST
U 106D, 0018.0038.0980.09F8.0000.11D2  ;2464  RC[0F]_K[.2]
U 11D2, 0000.003C.5D80.0800.0084.71D3  ;2465  SC_K[.7]  ; GENERATE A MASK USED TO
U 11D3, 0803.0010.6180.0800.0084.71D4  ;2466  ALU_MASK+1,D_ALU,SC_K[.F] ; TEST THE RESULT.
U 11D4, 0003.0010.01C0.0800.0000.11D5  ;2467  ALU_MASK+1,Q_ALU
U 11D5, 001D.0024.0180.0A88.0000.11D6  ;2468  ALU_D[ANDNOT]Q,R[1]_ALU
U 11D6, 0018.0038.8D80.0A98.0000.106E  ;2469  R[3]_K[.1F]
U 106E, 0000.003D.0180.0800.0000.1146  ;2470 =0 ERLOOP
U 106F, 0000.003C.0180.0800.0000.1070  ;2471  NOP
            ;2472 =0
U 1070, 0000.003D.0180.0800.0000.1150  ;2473 XRT1: CALL.J/FPINIT  ; INITIALIZE THE FPA.
U 1071, 0818.0038.8D80.0800.0000.1072  ;2474  D_K[.1F]  ; GENERATE ADDRESS OF FPA
U 1072, 0819.0015.0980.0800.0000.1156  ;2475 =0 ALU_D+K[.2],D_ALU,CALL.J/GENTRA ; UCODE.
U 1073, 0000.003C.5980.3E18.0082.11D7  ;2476  ID[ACC.2]_D,SC_R[3] ; GENERATE THE TEST DATA.
U 11D7, 0803.001C.0180.0A80.0000.11D8  ;2477  ALU_NOT.MASK,D_ALU,R[0]_ALU
U 11D8, 080D.0034.0180.0A08.0000.11D9  ;2478  ALU_D[AND]R[1],D_ALU ; GENERATE EXPECTED RESULT.
U 11D9, 0001.00FC.0380.09F0.0000.11DA  ;2479  TRAP.FPA,RC[0E]_D ; TRAP TO FPA LOCATION 21.
U 11DA, 0000.003C.0180.0A00.0000.11DB  ;2480  LAB_R[0]
U 11DB, 0003.007C.0180.0A80.0000.11DC  ;2481  R[0]_0,CPSYNC  ; TEST DATA LOADED HERE INTO LA.
U 11DC, 0000.003C.0180.0A00.0000.11DD  ;2482  LAB_R[0]  ; LA LOADED INTO XR HERE.
U 11DD, 0000.007C.0180.0A00.0000.11DE  ;2483  LAB_R[0],CPSYNC  ; LA AND LB LOADED WITH 0.
U 11DE, 0000.003C.01D8.0800.0000.11DF  ;2484  Q_ACC  ; READ THE RESULT.
U 11DF, 000D.2034.01C0.0A08.0000.11E0  ;2485  ALU_Q[AND]R[1],Q_ALU
U 11E0, 0001.203C.0180.09E8.0000.11E1  ;2486  RC[0D]_Q
U 11E1, 001D.0000.0180.0800.0010.11E2  ;2487  ALU_D-Q.CLK.UBCC ; RESULT CORRECT?
U 11E2, 0000.013C.0180.0800.0000.1074  ;2488  Z?
U 1074, 0000.003D.0180.0800.0000.1147  ;2489 =0 ERROR1  ; XR TEST FAILED.
```

```
U 1075, 0000,003C,0180,0A18,0010,11E3  ;2490  ALU_R[3],CLK.UBCC ; TEST DONE?
U 11E3, 0000,013C,0180,0800,0000,1076  ;2491  Z?
U 1076, 0018,0000,0580,0A98,0000,1070  ;2492 =0 R[3]_LA-K[.1],J/XRT1 ; NO SO CONTINUE.
U 1077, 0000,003C,0180,0800,0000,1078  ;2493  NOP
```

```
                ;2494  .PAGE "TEST 251 PR REGISTER, BITS <07:00>, AND AMX/PR, BITS <07:00>, TEST"
                ;2495 ;*****************************************************************
                ;2496 ;*
                ;2497 ; PR REGISTER, BITS <07:00>, AND AMX/PR, BITS <07:00>, TEST
                ;2498 ;
                ;2499 ; TEST DESCRIPTION
                ;2500 ; THIS IS A TEST OF BIT <07:00> OF THE PR REGISTER AND
                ;2501 ; THEIR INPUTS TO THE AMX. A 1 IS FLOATED ACROSS 0'S TO GENERATE
                ;2502 ; THE TEST DATA PATTERNS. NOTE THAT BITS <09:08> OF THE
                ;2503 ; PR AND THEIR INPUTS TO THE AMX ARE NOT TESTED HERE.
                ;2504 ; THESE TWO HIGH ORDER BITS ARE NOT DIRECTLY OBSERVABLE
                ;2505 ; THEY WILL BE TESTED LATER.
                ;2506 ;
                ;2507 ; FPA UCODE USED
                ;2508 ; LOCATION CONTENTS
                ;2509 ; 25:  BSC/R,SCR/CPU,EAC/LA,NAD/26,WAIT
                ;2510 ; 26:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/27
                ;2511 ; 27:  BSC/R,SCR/CPU,EAC/LDAB,NAD/2B,WAIT
                ;2512 ; 2B:  AMXC/PR,EALUC/A,BSC/NH,NAD/2B
                ;2513 ;
                ;2514 ; LOGIC DESCRIPTION
                ;2515 ;
                ;2516 ; ERROR DESCRIPTION
                ;2517 ; EXPECTED RESULT (BITS<14:07>)
                ;2518 ; GOT RESULT (BITS<14:07>)
                ;2519 ;
                ;2520 ; SYNC POINT DESCRIPTION
                ;2521 ;
                ;2522 ;--
                ;2523 ;*****************************************************************
                ;2524 =0
U 1078, 0000,003D,0180,0800,0000,1134  ;2525 PRT1: NEWTST
U 1079, 0018,0038,0980,09F0,0000,11E4  ;2526  RC[0E]_K[.2]
U 11E4, 0000,003C,5D80,0800,0084,71E5  ;2527  SC_K[.7]  ; GENERATE A MASK USED TO
U 11E5, 0803,0010,6180,0800,0084,71E6  ;2528  ALU_MASK*1,D_ALU,SC_K[.F] ; CALCULATE THE EXPECTED RESULT.
U 11E6, 0003,0010,01C0,0800,0000,11E7  ;2529  ALU_MASK*1,Q_ALU
U 11E7, 001D,0024,0180,0988,0000,11E8  ;2530  ALU_D[ANDNOT]Q,RC[1]_ALU
U 11E8, 0018,0038,8D80,0A98,0000,107A  ;2531  R[3]_K[.1F]  ; INITIALIZE A COUNTER.
U 107A, 0000,003D,0180,0800,0000,1146  ;2532 =0 ERLOOP
U 107B, 0000,003C,0180,0800,0000,107C  ;2533  NOP
                ;2534 =0
U 107C, 0000,003D,0180,0800,0000,1150  ;2535 PRT1A: CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 107D, 0818,0038,E980,0800,0000,107E  ;2536  D_K[.24]  ; GENERATE THE ADDRESS OF
U 107E, 0819,0015,0580,0800,0000,1156  ;2537 =0 ALU_D*K[.1],D_ALU,CALL,J/GENTRA ; THE FPA UCODE.
U 107F, 0000,003C,5980,3E18,0082,11E9  ;2538  ID[ACC.2]_D,SC_R[3] ; GET TEST DATA
U 11E9, 0803,001C,0180,0A80,0000,11EA  ;2539  ALU_NOT.MASK,D_ALU,R[0]_ALU
U 11EA, 080D,0034,0180,0A08,0000,11EB  ;2540  ALU_D[AND]R[1],D_ALU ; CALCULATE THE EXPECTED RESULT.
U 11EB, 0001,00FC,0380,09F0,0000,11EC  ;2541  TRAP,FPA,RC[0E]_D ; TRAP TO FPA LOCATION 25.
U 11EC, 0000,003C,0180,0A00,0000,11ED  ;2542  LAB_R[0]
U 11ED, 0003,007C,0180,0A80,0000,11EE  ;2543  R[0]_0,CPSYNC  ; TEST DATA LOADED INTO LA HERE.
U 11EE, 0000,003C,0180,0A00,0000,11EF  ;2544  LAB_R[0]  ; LA LOADED INTO PR HERE.
U 11EF, 0000,007C,0180,0A00,0000,11F0  ;2545  LAB_R[0],CPSYNC  ; LA AND LB LOADED WITH 0.
U 11F0, 0000,003C,01D8,0800,0000,11F1  ;2546  Q_ACC  ; GET THE RESULT.
U 11F1, 000D,2034,01C0,0AC8,0000,11F2  ;2547  ALU_Q[AND]R[1],Q_ALU
U 11F2, 0001,203C,0180,09E8,0000,11F3  ;2548  RC[0D]_Q
```

```
ZZ-ESKAR-V22  TEST 251 PR REGISTER, BITS <07:00>, AND AMX/PR, BITS <07:00>, TEST                    SEQ 2464
ESKAR5D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29                               Page   6

U 11F3, 001D.0000.0180.0800.0010.11F4  ;2549  ALU_D-Q,CLK.UBCC ; RESULT CORRECT?
U 11F4, 0000.013C.0180.0800.0000.1080  ;2550  Z?
U 1080, 0000.003D.0180.0800.0000.1147  ;2551 =0 ERROR1    ; PR TEST FAILED.
U 1081, 0000.003C.0180.0A18.0010.11F5  ;2552  ALU_R[3],CLK.UBCC ; TEST DONE?
U 11F5, 0000.013C.0180.0800.0000.1082  ;2553  Z?
U 1082, 0018.0000.0580.0A98.0000.107C  ;2554 =0 R[3]_LA-K[.1],J/PRT1A ; NO SO CONTINUE.
U 1083, 0000.003C.0180.0800.0000.1084  ;2555  NOP
```

```
                ;2556  .PAGE "TEST 252 EALU/K3FF TEST"
                ;2557 ;*****************************************************************
                ;2558 ;**
                ;2559 ; EALU/K3FF TEST
                ;2560 ;
                ;2561 ; TEST DESCRIPTION
                ;2562 ; THIS IS A TEST OF THE EALU FUNCTION WHICH DRIVES EALU<9:0>
                ;2563 ; TO 3FF. NOTE THAT ONLY EALU<7:0> CAN BE OBSERVED
                ;2564 ; DIRECTLY. EALU/K3FF BITS <9:8> WILL BE TESTED LATER.
                ;2565 ; TWO TESTS ARE PERFORMED FIRST EALUC/K3FF IS SELECTED AND THE
                ;2566 ; BSC/NH FUNCTION USED, THE RESULT SHOULD BE THAT 0'S ARE PLACED
                ;2567 ; ON BUS A <14:07>. THEN EALUC/K3FF IS SELECTED AND THE RESULT
                ;2568 ; PUT IN THE XR FROM WHICH IT IS READ AND CHECKED, HERE BUS A <14:07>
                ;2569 ; WILL BE ALL 1'S.
                ;2570 ;
                ;2571 ; FPA UCODE USED
                ;2572 ; LOCATION CONTENTS
                ;2573 ; 1B:  EALUC/K3FF,EAC/LDXR,BSC/NH,NAD/63
                ;2574 ; 63:  BSC/R,SCR/CPU,EAC/LDAB,NAD/64,WAIT
                ;2575 ; 64:  BMXC/XR,AMXC/LA,EALUC/A+B,BSC/NH,NAD/64
                ;2576 ;
                ;2577 ; LOGIC DESCRIPTION
                ;2578 ;
                ;2579 ; ERROR DESCRIPTION
                ;2580 ; EXPECTED OUTPUT
                ;2581 ; GOT OUTPUT
                ;2582 ;
                ;2583 ; SYNC POINT DESCRIPTION
                ;2584 ;
                ;2585 ;--
                ;2586 ;*****************************************************************
                ;2587 =0
U 1084, 0000.003D,0180.0800,0000.1134  ;2588 EALUK: NEWTST
U 1085, 0018.0038,0980.09F8,0000.1086  ;2589  RC[0F]_K[.2]
U 1086, 0000.003D,0180.0800,0000.1150  ;2590 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1087, 0000.003C,5D80.0800,0084.71F6  ;2591  SC_K[.7]  ; GENERATE EXPECTED RESULT.
U 11F6, 0803.0010,6180.0800,0084.71F7  ;2592  ALU_MASK+1,D_ALU,SC_K[.F]
U 11F7, 0003.0010,01C0.0800,0000.11F8  ;2593  ALU_MASK+1,Q_ALU
U 11F8, 001D.0024,0180.09F0,0000.11F9  ;2594  ALU_D[ANDNOT]Q,RC[0E]_ALU
U 11F9, 0818.0038,E580.0800,0000.1088  ;2595  D_K[.1A]  ; GENERATE ADDRESS OF FPA UCODE.
U 1088, 0819.0015,0580.0800,0000.1156  ;2596 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 1089, 0000.003C,5980.3C00,0000.11FA  ;2597  ID[ACC.2]_D
U 11FA, 0003.00FC,0380.0A80,0000.11FB  ;2598  TRAP.FPA,R[0]_0
U 11FB, 0000.003C,01D8.0800,0000.11FC  ;2599  Q_ACC  ; GET RESULT
U 11FC, 0000.003C,0180.0A00,0000.11FD  ;2600  LAB_R[0]
U 11FD, 0000.007C,0180.0A00,0000.11FE  ;2601  LAB_R[0],CPSYNC
U 11FE, 0A00.003C,0180.0800,0000.11FF  ;2602  D_ACC
U 11FF, 0001.203C,0180.0AA8,0000.1200  ;2603  R[5]_Q
U 1200, 0010.0038,01C0.0970,0000.1201  ;2604  Q_RC[0E]
U 1201, 081D.2034,0180.09E8,0000.1202  ;2605  ALU_Q[AND]D,D_ALU,RC[0D]_ALU
U 1202, 001D.0000,0180.0800,0010.1203  ;2606  ALU_D-Q,CLK.UBCC
U 1203, 0000.013C,0180.0800,0000.108A  ;2607  Z?
U 108A, 0000.003D,0180.0800,0000.1147  ;2608 =0 ERROR1  ; EALU OUTPUT NOT 00007F80
U 108B, 0003.003C,0180.09F0,0000.1204  ;2609  RC[0E]_0
U 1204, 0800.003C,0180.0A28,0000.1205  ;2610  D_R[5]
```

```
ZZ-ESKAR-V22  TEST 252 EALU/K3FF TEST                                                    SEQ 2466
ESKAR5D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29                   Page   7

U 1205, 001D.2034.0180.09E8.0010.1206  ;2611  ALU_Q[AND]D,RC[0D]_ALU,CLK.UBCC
U 1206, 0000.013C.0180.0800.0000.108C  ;2612  Z?
U 108C, 0000.003D.0180.0800.0000.1147  ;2613 =0 ERROR1    ; BUS A <14:07> WAS NOT 0.
U 108D, 0000.003C.0180.0800.0000.108E  ;2614  NOP
```

```
            ;2615  .PAGE "TEST 253 LA LOAD FUNCTION TEST"
            ;2616 ;*****************************************************************
            ;2617 ;++
            ;2618 ; LA LOAD FUNCTION TEST
            ;2619 ;
            ;2620 ; TEST DESCRIPTION
            ;2621 ; THIS IS A TEST OF THE LA REGISTER LOAD FUNCTION. IT INSURES THAT
            ;2622 ; LA IS NOT LOADED WHEN IT SHOULD NOT BE.
            ;2623 ;
            ;2624 ; FPA UCODE USED
            ;2625 ; LOCATION CONTENTS
            ;2626 ; 70:  BSC/R,SCR/CPU,EAC/LDA,NAD/71,WAIT
            ;2627 ; 71:  BSC/R,SCR/CPU,EALUC/K3FF,EAC/PR.XR.LB,NAD/72,WAIT
            ;2628 ; 72:  AMXC/LA,EALUC/A,BSC/NH,NAD/PSTALL
            ;2629 ;
            ;2630 ; LOGIC DESCRIPTION
            ;2631 ;
            ;2632 ; ERROR DESCRIPTION
            ;2633 ; EXPECTED DATA FROM LA (BITS <14:07>)
            ;2634 ; GOT DATA FROM LA (BITS <14:07>)
            ;2635 ;
            ;2636 ; SYNC POINT DESCRIPTION
            ;2637 ;
            ;2638 ;--
            ;2639 ;*****************************************************************
            ;2640 =0
U 108E, 0000.003D,0180.0800.0000.1134  ;2641 LALDT: NEWTST
U 108F, 0018.0038,0980.09F8.0000.1207  ;2642  RC[0F]_K[.2]
U 1207, 0000.003C,5D80.0800.0084.7208  ;2643  SC_K[.7]
U 1208, 0803.0010,6180.0800.0084.7209  ;2644  ALU_MASK+1,D_ALU,SC_K[.F] ; GENERATE A MASK USED TO TEST
U 1209, 0003.0010,01C0.0800.0000.120A  ;2645  ALU_MASK+1,Q_ALU ; THE RESULT
U 120A, 001D.0024,01C0.0A90.0000.120B  ;2646  ALU_D[ANDNOT]Q,Q_ALU,R[2]_ALU ; R[2]=00007F80
U 120B, 0818.0038,2580.0800.0000.120C  ;2647  D_K[.A0]  ; GENERATE THE TEST DATA.
U 120C, 0819.0030,F5F8.0800.0000.120D  ;2648  ALU_D.OR.K[.A],D_ALU,Q_0
U 120D, 0000.003C,5D80.0800.0084.720E  ;2649  SC_K[.7]
U 120E, 0D03.003C,0180.0A80.0000.120F  ;2650  D_DAL.SC,R[0]_0
U 120F, 0001.003C,0180.0A88.0000.1210  ;2651  R[1]_D   ; R[1]=00005500
U 1210, 0001.003C,0180.09F0.0000.1090  ;2652  RC[0E]_D  ; RC[0E]=00005500
U 1090, 0000.003D,0180.0800.0000.1150  ;2653 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 1091, 0818.0038,9D80.0800.0000.1092  ;2654  D_K[.7C]  ; GENERATE THE ADDRESS OF THE FPA
U 1092, 0819.0021,8580.0800.0000.1156  ;2655 =0 ALU_D.XOR.K[.C],D_ALU,CALL,J/GENTRA ; UCODE
U 1093, 0000.003C,5980.3C00.0000.1211  ;2656  ID[ACC.2]_D
U 1211, 0000.00FC,0380.0800.0000.1212  ;2657  TRAP.FPA  ; TRAP TO FPA 70.
U 1212, 0800.003C,0180.0A08.0000.1213  ;2658  D_R[1]   ; LA LOADED HERE IN FPA.
U 1213, 0000.007C,0180.0A08.0000.1214  ;2659  LAB_R[1],CPSYNC
U 1214, 0000.003C,0180.0A10.0000.1215  ;2660  LAB_R[2]
U 1215, 0000.007C,0180.0A10.0000.1216  ;2661  LAB_R[2],CPSYNC
U 1216, 0000.003C,01D8.0800.0000.1217  ;2662  Q_ACC   ; GET THE RESULT
U 1217, 000D.2034,01C0.0A10.0000.1218  ;2663  ALU_Q[AND]R[2],Q_ALU
U 1218, 0001.203C,0180.09E8.0000.1219  ;2664  RC[0D]_Q
U 1219, 001D.0000,0180.0800.0010.121A  ;2665  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
U 121A, 0000.013C,0180.0800.0000.1094  ;2666  Z?
U 1094, 0000.003D,0180.0800.0000.1147  ;2667 =0 ERROR1   ; LA DATA NOT CORRECT.
U 1095, 0000.003C,0180.0800.0000.1096  ;2668  NOP
```

```
          ;2669  .PAGE "TEST 254 LB LOAD FUNCTION TEST"
          ;2670 ;*****************************************************************
          ;2671 ;**
          ;2672 ; LB LOAD FUNCTION TEST
          ;2673 ;
          ;2674 ; TEST DESCRIPTION
          ;2675 ; THIS IS A TEST OF THE LOAD SIGNAL ON THE LB REGISTER.
          ;2676 ; THIS TEST INSURE THAT IT IS ASSERTED AND DEASSERTED AT
          ;2677 ; APPROPRIATE TIMES.
          ;2678 ;
          ;2679 ; FPA UCODE USED
          ;2680 ; LOCATION CONTENTS
          ;2681 ; 73:  BSC/R,SCR/CPU,EAC/LDB,NAD/74,WAIT
          ;2682 ; 74:  BSC/R,SCR/CPU,EALUC/K3FF,EAC/PR.XR.LA,NAD/75,WAIT
          ;2683 ; 75:  AMXC/LA,BMXC/LB,EALUC/A+B,BSC/NH,NAD/PSTALL
          ;2684 ;
          ;2685 ; LOGIC DESCRIPTION
          ;2686 ;
          ;2687 ; ERROR DESCRIPTION
          ;2688 ; EXPECTED DATA FROM LB (BITS <14:07>)
          ;2689 ; GOT DATA FROM LB (BITS <14:07>)
          ;2690 ;
          ;2691 ; SYNC POINT DESCRIPTION
          ;2692 ;
          ;2693 ;--
          ;2694 ;*****************************************************************
          ;2695 =0
U 1096, 0000.003D.0180.0800.0000.1134  ;2696 LBLDT: NEWTST
U 1097, 0018.0038.0980.09F8.0000.121B  ;2697  RC[0F]_K[.2]
U 121B, 0000.003C.5D80.0800.0084.721C  ;2698  SC_K[.7]
U 121C, 0803.0010.6180.0800.0084.721D  ;2699  ALU_MASK+1,D_ALU,SC_K[.F] ; GENERATE A MASK USED TO TEST
U 121D, 0003.0010.01C0.0800.0000.121E  ;2700  ALU_MASK+1,Q_ALU ; THE RESULT
U 121E, 001D.0024.01C0.0A90.0000.121F  ;2701  ALU_D[ANDNOT]Q,Q_ALU,R[2]_ALU ; R[2]=00007F80
U 121F, 0818.0038.2580.0800.0000.1220  ;2702  D_K[.A0]  ; GENERATE THE TEST DATA.
U 1220, 0819.0030.F5F8.0800.0000.1221  ;2703  ALU_D.OR.K[.A],D_ALU,Q_0
U 1221, 0000.003C.D580.0800.0084.7222  ;2704  SC_K[.6]
U 1222, 0D03.003C.0180.0A80.0000.1223  ;2705  D_DAL.SC,R[0]_0
U 1223, 0001.003C.0180.0A88.0000.1224  ;2706  R[1]_D   ; R[1]=00002A80
U 1224, 0001.003C.0180.09F0.0000.1098  ;2707  RC[0E]_D  ; RC[0E]=00002AB0
U 1098, 0000.003D.0180.0800.0000.1150  ;2708 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 1099, 0818.0038.9D80.0800.0000.109A  ;2709  D_K[.7C]  ; GENERATE THE ADDRESS OF THE FPA
U 109A, 0819.0021.6180.0800.0000.1156  ;2710 =0 ALU_D.XOR.K[.F],D_ALU,CALL,J/GENTRA ; UCODE
U 109B, 0000.003C.5980.3C00.0000.1225  ;2711  ID[ACC.2]_D
U 1225, 0000.00FC.0380.0800.0000.1226  ;2712  TRAP.FPA  ; TRAP TO FPA 73.
U 1226, 0800.003C.0180.0A08.0000.1227  ;2713  D_R[1]   ; LB LOADED HERE IN FPA.
U 1227, 0000.007C.0180.0A08.0000.1228  ;2714  LAB_R[1],CPSYNC
U 1228, 0000.003C.0180.0A00.0000.1229  ;2715  LAB_R[0]
U 1229, 0000.007C.0180.0A00.0000.122A  ;2716  LAB_R[0],CPSYNC
U 122A, 0000.003C.01D8.0800.0000.122B  ;2717  Q_ACC   ; GET THE RESULT
U 122B, 000D.2034.01C0.0A10.0000.122C  ;2718  ALU_Q[AND]R[2],Q_ALU
U 122C, 0001.203C.0180.09E8.0000.122D  ;2719  RC[0D]_Q
U 122D, 001D.0000.0180.0800.0010.122E  ;2720  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
U 122E, 0000.013C.0180.0800.0000.109C  ;2721  Z?
U 109C, 0000.003D.0180.0800.0000.1147  ;2722 =0 ERROR1   ; LB DATA NOT CORRECT.
U 109D, 0000.003C.0180.0800.0000.109E  ;2723  NOP
```

```
            :2724  .PAGE "TEST 255 PR LOAD FUNCTION TEST"
            :2725 ;*****************************************************************
            :2726 ;**
            :2727 ; PR LOAD FUNCTION TEST
            :2728 ;
            :2729 ; TEST DESCRIPTION
            :2730 ; THIS IS A TEST OF THE PR REGISTER LOAD SIGNAL.
            :2731 ; IT INSURES THAT THE PR REGISTER IS LOADED WHEN IT
            :2732 ; IS SUPPOSED TO BE LOADED AND ALSO NOT LOADED WHEN
            :2733 ; IT SHOULD NOT BE LOADED.
            :2734 ;
            :2735 ; FPA UCODE USED
            :2736 ; LOCATION CONTENTS
            :2737 ; 76:  BSC/R,SCR/CPU,EAC/LDA,NAD/77,WAIT
            :2738 ; 77:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/78
            :2739 ; 78:  BSC/R,SCR/CPU,EALUC/K3FF,EAC/XR.AB,NAD/79,WAIT
            :2740 ; 79:  AMXC/PR,EALUC/A,BSC/NH,NAD/PSTALL
            :2741 ;
            :2742 ; LOGIC DESCRIPTION
            :2743 ;
            :2744 ; ERROR DESCRIPTION
            :2745 ; EXPECTED DATA FROM PR (BITS <14:07>)
            :2746 ; GOT DATA FROM PR (BITS <14:07>)
            :2747 ;
            :2748 ; SYNC POINT DESCRIPTION
            :2749 ;
            :2750 ;--
            :2751 ;*****************************************************************
            :2752 =0
U 109E, 0000.003D.0180.0800.0000.1134  :2753 PRLDT: NEWTST
U 109F, 001B.0038.0980.09F8.0000.122F  :2754  RC[0F]_K[.2]
U 122F, 0000.003C.5D80.0800.0084.7230  :2755  SC_K[.7]
U 1230, 0803.0010.6180.0800.0084.7231  :2756  ALU_MASK*1,D_ALU,SC_K[.F] ; GENERATE A MASK USED TO TEST
U 1231, 0003.0010.01C0.0800.0000.1232  :2757  ALU_MASK*1,Q_ALU ; THE RESULT
U 1232, 001D.0024.01C0.0A90.0000.1233  :2758  ALU_D[ANDNOT]Q,Q_ALU,R[2]_ALU ; R[2]=00007F80
U 1233, 0818.0038.2580.0800.0000.1234  :2759  D_K[.A0]  ; GENERATE THE TEST DATA.
U 1234, 0819.0030.F5F8.0800.0000.1235  :2760  ALU_D.OR.K[.A],D_ALU,Q_0
U 1235, 0000.003C.5D80.0800.0084.7236  :2761  SC_K[.7]
U 1236, 0D03.003C.0180.0A80.0000.1237  :2762  D_DAL.SC,R[0]_0
U 1237, 0001.003C.0180.0A88.0000.1238  :2763  R[1]_D   ; R[1]=00005500
U 1238, 0001.003C.0180.09F0.0000.10A0  :2764  RC[0E]_D  ; RC[0E]=00005500
U 10A0, 0000.003D.0180.0800.0000.1150  :2765 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 10A1, 0818.0038.F980.0800.0000.10A2  :2766  D_K[.7E]  ; GENERATE THE ADDRESS OF THE FPA
U 10A2, 0819.0001.0180.0800.0000.1156  :2767 =0 ALU_D-K[.8],D_ALU,CALL,J/GENTRA ; UCODE
U 10A3, 0000.003C.5980.3C00.0000.1239  :2768  ID[ACC.2]_D
U 1239, 0000.00FC.0380.0800.0000.123A  :2769  TRAP.FPA  ; TRAP TO FPA 76.
U 123A, 0000.003C.0180.0A08.0000.123B  :2770  LAB_R[1]  ; LA LOADED IN THE FPA HERE.
U 123B, 0000.007C.0180.0A08.0000.123C  :2771  LAB_R[1],CPSYNC
U 123C, 0000.003C.0180.0A10.0000.123D  :2772  LAB_R[2]  ; PR LOADED HERE.
U 123D, 0000.007C.0180.0A10.0000.123E  :2773  LAB_R[2],CPSYNC
U 123E, 0000.003C.01D8.0800.0000.123F  :2774  Q_ACC   ; GET THE RESULT
U 123F, 000D.2034.01C0.0A10.0000.1240  :2775  ALU_Q[AND]R[2],Q_ALU
U 1240, 0810.0038.0180.0970.0000.1241  :2776  D_RC[0E]
U 1241, 0001.203C.0180.09E8.0000.1242  :2777  RC[0D]_Q
U 1242, 001D.0000.0180.0800.0010.1243  :2778  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
```

```
ZZ-ESKAR-V22  TEST 255 PR LOAD FUNCTION TEST                                    SEQ 2470
ESKAR5D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29          Page   7

U 1243. 0000.013C.0180.0800.0000.10A4  ;2779  Z?
U 10A4. 0000.003D.0180.0800.0000.1147  ;2780 =0 ERROR1    ; PR DATA NOT CORRECT.
U 10A5. 0000.003C.0180.0800.0000.10A6  ;2781  NOP
```

```
                ;2782  .PAGE "TEST 256 XR LOAD FUNCTION TEST"
                ;2783 ;*****************************************************************
                ;2784 ;**
                ;2785 ; XR LOAD FUNCTION TEST
                ;2786 ;
                ;2787 ; TEST DESCRIPTION
                ;2788 ; THIS IS A TEST OF THE XR REGISTER LOAD FUNCTION. THE TEST
                ;2789 ; INSURES THAT THE REGISTER IS LOADED AND NOT LOADED UNDER
                ;2790 ; THE APPROPRIATE CONDITIONS.
                ;2791 ;
                ;2792 ; FPA UCODE USED
                ;2793 ; LOCATION CONTENTS
                ;2794 ; 7A:  BSC/R,SCR/CPU,EAC/LDLA,NAD/7B,WAIT
                ;2795 ; 7B:  AMXC/LA,EALUC/A,EAC/LDXR,NAD/7C
                ;2796 ; 7C:  BSC/R,SCR/CPU,EALUC/K3FF,EAC/PR.AB,NAD/7D,WAIT
                ;2797 ; 7D:  AMXC/LA,BMXC/XR,EALUC/A+B,BSC/NH,NAD/PSTALL
                ;2798 ;
                ;2799 ; LOGIC DESCRIPTION
                ;2800 ;
                ;2801 ; ERROR DESCRIPTION
                ;2802 ; EXPECTED DATA FROM XR (BITS <14:07>)
                ;2803 ; GOT DATA FROM XR (BITS <14:07>)
                ;2804 ;
                ;2805 ; SYNC POINT DESCRIPTION
                ;2806 ;
                ;2807 ;--
                ;2808 ;*****************************************************************
                ;2809 =0
U 10A6. 0000.003D.0180.0800.0000.1134   ;2810 XRLDT: NEWTST
U 10A7. 0018.0038.0980.09F8.0000.1244   ;2811  RC[0F]_K[.2]
U 1244. 0000.003C.5D80.0800.0084.7245   ;2812  SC_K[.7]
U 1245. 0803.0010.6180.0800.0084.7246   ;2813  ALU_MASK+1,D_ALU,SC_K[.F] ; GENERATE A MASK USED TO TEST
U 1246. 0003.0010.01C0.0800.0000.1247   ;2814  ALU_MASK+1,Q_ALU ; THE RESULT
U 1247. 001D.0024.01C0.0A90.0000.1248   ;2815  ALU_D[ANDNOT]Q,Q_ALU,R[2]_ALU ; R[2]=00007F80
U 1248. 0818.0038.2580.0800.0000.1249   ;2816  D_K[.A0]  ; GENERATE THE TEST DATA.
U 1249. 0819.0030.F5F8.0800.0000.124A   ;2817  ALU_D.OR.K[.A],D_ALU,Q_0
U 124A. 0000.003C.D580.0800.0084.724B   ;2818  SC_K[.6]
U 124B. 0D03.003C.0180.0A80.0000.124C   ;2819  D_DAL.SC,R[0]_0
U 124C. 0001.003C.0180.0A88.0000.124D   ;2820  R[1]_D   ; R[1]=00002A80
U 124D. 0001.003C.0180.09F0.0000.10A8   ;2821  RC[0E]_D  ; RC[0E]=00002A80
U 10A8. 0000.003D.0180.0800.0000.1150   ;2822 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 10A9. 0818.0038.9D80.0800.0000.10AA   ;2823  D_K[.7C]  ; GENERATE THE ADDRESS OF THE FPA
U 10AA. 0819.0001.0980.0800.0000.1156   ;2824 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; UCODE
U 10AB. 0000.003C.5980.3C00.0000.124E   ;2825  ID[ACC.2]_D
U 124E. 0000.00FC.0380.0800.0000.124F   ;2826  TRAP.FPA  ; TRAP TO FPA 7A.
U 124F. 0800.003C.0180.0A08.0000.1250   ;2827  D_R[1]   ; LA LOADED HERE IN FPA.
U 1250. 0000.007C.0180.0A08.0000.1251   ;2828  LAB_R[1],CPSYNC
U 1251. 0000.003C.0180.0A00.0000.1252   ;2829  LAB_R[0]  ; XR LOADED HERE
U 1252. 0000.007C.0180.0A00.0000.1253   ;2830  LAB_R[0],CPSYNC
U 1253. 0000.003C.01D8.0800.0000.1254   ;2831  Q_ACC   ; GET THE RESULT
U 1254. 000D.2034.01C0.0A10.0000.1255   ;2832  ALU_Q[AND]R[2],Q_ALU
U 1255. 0001.203C.0180.09E8.0000.1256   ;2833  RC[0D]_Q
U 1256. 001D.0000.0180.0800.0010.1257   ;2834  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
U 1257. 0000.013C.0180.0800.0000.10AC   ;2835  Z?
U 10AC. 0000.003D.0180.0800.0000.1147   ;2836 =0 ERROR1   ; XR DATA NOT CORRECT.
```

```
U 10AD, 0000,003C,0180,0800,0000,10AE  ;2837  NOP
```

```
                ;2838  .PAGE "TEST 257 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 1"
                ;2839  ;*****************************************************************
                ;2840  ;**
                ;2841  ; PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 1
                ;2842  ;
                ;2843  ; TEST DESCRIPTION
                ;2844  ; THIS TEST PUTS 0 ON THE EALU OUTPUT AND LOADS THE PSL
                ;2845  ; CONDITION CODES FROM THE FPA. THIS SHOULD CAUSE JUST THE
                ;2846  ; Z BIT TO SET. THEN ZERO IS LOADED INTO THE PR
                ;2847  ; AND THE PR IS GATED THRU THE EALU TO LOAD AND CHECK
                ;2848  ; THE CONDITION CODES AGAIN.
                ;2849  ;
                ;2850  ; FPA UCODE USED
                ;2851  ; LOCATION CONTENTS
                ;2852  ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;2853  ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;2854  ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;2855  ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;2856  ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;2857  ; 6F:  BEN/7,NAD/0
                ;2858  ;
                ;2859  ; LOGIC DESCRIPTION
                ;2860  ;
                ;2861  ; ERROR DESCRIPTION
                ;2862  ; EXPECTED PSL CONDITION CODES, Z V C BITS <2:0>
                ;2863  ; GOT PSL CONDITION CODES, Z V C BITS <2:0>
                ;2864  ; EXPECTED EALU OUTPUT <09:00>
                ;2865  ;
                ;2866  ; SYNC POINT DESCRIPTION
                ;2867  ;
                ;2868  ;--
                ;2869  ;*****************************************************************
                ;2870  =0
U 10AE.  0000,003D,0180,0800,0000,1134  ;2871 CCT1: NEWTST
U 10AF.  001B,0038,0D80,09F8,0000,1258  ;2872  RC[0F]_K[.3]
U 1258.  0003,003C,0180,09E0,0000,1259  ;2873  RC[0C]_0
U 1259.  001B,0038,1180,09F0,0000,10B0  ;2874  RC[0E]_K[.4]
U 10B0.  0000,003D,0180,0800,0000,1150  ;2875 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 10B1.  081B,0038,7980,0800,0000,10B2  ;2876  D_K[.30]  ; ESTABLISH THE ADDRESS OF
U 10B2.  0819,0001,0980,0800,0000,1156  ;2877 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; THE FPA UCODE USED.
U 10B3.  0003,003C,5980,3E80,0000,125A  ;2878  ID[ACC.2]_D,R[0]_0
U 125A.  081B,0038,0D80,0800,0000,125B  ;2879  D_K[.3]
U 125B.  0000,00FC,3F80,3C00,0000,125C  ;2880  TRAP.FPA,ID[PSL]_D ; TRAP THE FPA TO 2E. PSL=3.
U 125C.  0000,003C,0180,0A00,0000,125D  ;2881  LAB_R[0]  ; AT FPA 2E.
U 125D.  0000,007C,0180,0A00,0000,125E  ;2882  LAB_R[0],CPSYNC
U 125E.  0000,003C,0180,0A00,0000,125F  ;2883  LAB_R[0]  ; AT FPA 6E
U 125F.  0000,003C,0180,0A00,1800,1260  ;2884  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 1260.  0000,003C,3DF0,2E00,0000,1261  ;2885  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 1261.  0C00,003C,3D80,3E00,0000,1262  ;2886  LAB_R[0],ID[PSL]_D,D_Q ; AT FPA 2C
U 1262.  0000,003C,0180,0A00,0000,1263  ;2887  LAB_R[0]  ; AT FPA 2D
U 1263.  0000,003C,0180,0A00,1800,1264  ;2888  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 1264.  0000,003C,3DF0,2E00,0000,1265  ;2889  LAB_R[0],Q_ID[PSL]
U 1265.  0819,0034,5D80,09E8,0000,1266  ;2890  ALU_D[AND]K[.7],D_ALU,RC[0D]_ALU
U 1266.  0019,0000,1180,0800,0010,1267  ;2891  ALU_D-K[.4],CLK.UBCC ; CHECK FIRST RESULT.
U 1267.  0000,013C,0180,0800,0000,10B4  ;2892  Z?
```

```
U 10B4, 0000,003D,0180,0800,0000,1147  ;2893 =0 ERROR1   ; LOAD ACC.CC FAILED OR EALU
        ;2894     ; BITS <9:8> ARE INCORRECT.
U 10B5, 0019,2034,5DC0,09E8,0000,1268  ;2895  ALU_Q[AND]K[.7],Q_ALU,RC[0D]_ALU
U 1268, 0019,2000,1180,0800,0010,1269  ;2896  ALU_Q-K[.4],CLK.UBCC ; CHECK SECOND RESULT.
U 1269, 0000,013C,0180,0800,0000,10B6  ;2897  Z?
U 10B6, 0000,003D,0180,0800,0000,1147  ;2898 =0 ERROR1   ; PR BITS <9:8> OR AMX/PR BITS
        ;2899     ; <9:8> FAILED.
U 10B7, 0000,003C,0180,0800,0000,10B8  ;2900  NOP
```

```
                ;2901   .PAGE "TEST 258 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 2"
                ;2902   ;*****************************************************************
                ;2903   ;**
                ;2904   ; PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 2
                ;2905   ;
                ;2906   ; TEST DESCRIPTION
                ;2907   ; THIS TEST PUTS 100 ON THE OUTPUT OF THE EALU AND LOADS THE
                ;2908   ; PSL CONDITION CODES FROM THE FPA. THIS SHOULD CAUSE THE V
                ;2909   ; BIT TO SET. THEN 100 IS LOADED INTO THE PR AND THE PR IS GATED
                ;2910   ; THRU THE EALU TO LOAD AND CHECK THE CONDITION CODES AGAIN.
                ;2911   ;
                ;2912   ; FPA UCODE USED
                ;2913   ; LOCATION CONTENTS
                ;2914   ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;2915   ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;2916   ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;2917   ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;2918   ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;2919   ; 6F:  BEN/7,NAD/0
                ;2920   ;
                ;2921   ; LOGIC DESCRIPTION
                ;2922   ;
                ;2923   ; ERROR DESCRIPTION
                ;2924   ; EXPECTED PSL CONDITION CODES, Z V C BITS <2:0>
                ;2925   ; GOT PSL CONDITION CODES, Z V C BITS <2:0>
                ;2926   ; EXPECTED EALU OUTPUT <9:0>
                ;2927   ;
                ;2928   ; SYNC POINT DESCRIPTION
                ;2929   ;
                ;2930   ;--
                ;2931   ;*****************************************************************
                ;2932   =0
U 10B8, 0000.003D.0180.0800.0000.1134   ;2933 CCT2: NEWTST
U 10B9, 0018.0038.0D80.09F8.0000.126A   ;2934  RC[0F]_K[.3]
U 126A, 0000.003C.0180.0800.0084.726B   ;2935  SC_K[.8]
U 126B, 0003.001C.0180.09E0.0000.126C   ;2936  ALU_NOT.MASK,RC[0C]_ALU
U 126C, 0018.0038.0980.09F0.0000.10BA   ;2937  RC[0E]_K[.2]
U 10BA, 0000.003D.0180.0800.0000.1150   ;2938 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 10BB, 0818.0038.7980.0800.0000.10BC   ;2939  D_K[.30]  ; GENERATE THE ADDRESS OF THE
U 10BC, 0819.0001.0980.0800.0000.1156   ;2940 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 10BD, 0003.003C.5980.3E80.0000.126D   ;2941  ID[ACC.2]_D,R[0]_0
U 126D, 0058.0038.4580.0A88.0000.126E   ;2942  ALU_K[.8000],R[1]_ALU.RIGHT,SI/ZERO ; EXPONENT = 80
U 126E, 0818.0038.1180.0800.0000.126F   ;2943  D_K[.4]
U 126F, 0819.0014.05C0.0800.0000.1270   ;2944  ALU_D+K[.1],Q_ALU,D_ALU
U 1270, 0000.00FC.3F80.3C00.0000.1271   ;2945  TRAP,FPA,ID[PSL]_D ; TRAP TO FPA 2E
U 1271, 0000.003C.0180.0A08.0000.1272   ;2946  LAB_R[1]  ; AT FPA 2E
U 1272, 0000.007C.0180.0A08.0000.1273   ;2947  LAB_R[1],CPSYNC
U 1273, 0000.003C.0180.0A08.0000.1274   ;2948  LAB_R[1]  ; AT FPA 6E
U 1274, 0000.003C.0180.0A08.0000.1275   ;2949  LAB_R[1]  ; AT FPA 2C
U 1275, 0000.003C.0180.0A00.0000.1276   ;2950  LAB_R[0]  ; AT FPA 2D
U 1276, 0000.003C.0180.0A00.1800.1277   ;2951  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 1277, 0000.003C.3DF0.2E00.0000.1278   ;2952  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 1278, 0C00.003C.3D80.3E00.0000.1279   ;2953  LAB_R[0],ID[PSL]_D,D_Q ; AT FPA 2C
U 1279, 0000.003C.0180.0A00.0000.127A   ;2954  LAB_R[0]  ; AT FPA 2D
U 127A, 0000.003C.0180.0A00.1800.127B   ;2955  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
```

```
ZZ-ESKAR-V22  TEST 258 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 2                    SEQ 2476
ESKAR5D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29                     Page   8

U 127B, 0000.003C.3DF0.2E00.0000.127C   ;2956  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 127C, 0819.0034.5D80.09E8.0000.127D   ;2957  ALU_D[AND]K[.7],RC[0D]_ALU,D_ALU
U 127D, 0019.0000.0980.0800.0010.127E   ;2958  ALU_D-K[.2],CLK.UBCC ; CHECK FIRST RESULT
U 127E, 0000.013C.0180.0800.0000.10BE   ;2959  Z?
U 10BE, 0000.003D.0180.0800.0000.1147   ;2960 =0 ERROR1    ; EALU BITS <9:8> FAILED OR
          ;2961     ; LOAD.ACC.CC FAILED.
U 10BF, 0019.2034.5DC0.09E8.0000.127F   ;2962  ALU_Q[AND]K[.7],Q_ALU,RC[0D]_ALU
U 127F, 0019.2000.0980.0800.0010.1280   ;2963  ALU_Q-K[.2],CLK.UBCC ; CHECK SECOND RESULT
U 1280, 0000.013C.0180.0800.0000.10C0   ;2964  Z?
U 10C0, 0000.003D.0180.0800.0000.1147   ;2965 =0 ERROR1    ; PR <9:8> OR AMX/PR <9:8> FAILED.
U 10C1, 0000.003C.0180.0800.0000.10C2   ;2966  NOP
```

```
                ;2967  .PAGE "TEST 259 PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 3"
                ;2968 ;*****************************************************************
                ;2969 ;**
                ;2970 ; PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 3
                ;2971 ;
                ;2972 ; TEST DESCRIPTION
                ;2973 ; THIS TEST PUTS 300 ON THE EALU OUTPUT AND LOADS THE PSL
                ;2974 ; CONDITION CODES FROM THE FPA. THIS SHOULD CAUSE JUST THE
                ;2975 ; Z BIT TO SET. THEN 300 IS LOADED INTO THE PR REGISTER, THE
                ;2976 ; PR IS GATED THRU THE EALU AND THE CONDITION CODES ARE
                ;2977 ; LOADED AND CHECKED AGAIN.
                ;2978 ;
                ;2979 ; FPA UCODE USED
                ;2980 ; LOCATION CONTENTS
                ;2981 ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;2982 ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;2983 ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;2984 ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;2985 ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;2986 ; 6F:  BEN/7,NAD/0
                ;2987 ;
                ;2988 ; LOGIC DESCRIPTION
                ;2989 ;
                ;2990 ; ERROR DESCRIPTION
                ;2991 ; EXPECTED PSL CONDITION CODES, Z V C BITS <2:0>
                ;2992 ; GOT PSL CONDITION CODES, Z V C BITS <2:0>
                ;2993 ; EXPECTED EALU OUTPUT <09:00>
                ;2994 ;
                ;2995 ; SYNC POINT DESCRIPTION
                ;2996 ;
                ;2997 ;--
                ;2998 ;*****************************************************************
                ;2999 =0
U 10C2. 0000.003D.0180.0800.0000.1134  ;3000 CCT3: NEWTST
U 10C3. 0018.0038.0D80.09F8.0000.1281  ;3001  RC[0F]_K[.3]
U 1281. 0000.003C.0180.0800.0084.7282  ;3002  SC_K[.8]
U 1282. 0803.001C.0580.0800.0084.9283  ;3003  ALU_NOT.MASK,D_ALU,SC_SC+K[.1]
U 1283. 0001.001C.0180.09E0.0000.1284  ;3004  ALU_D.ORNOT.MASK,RC[0C]_ALU
U 1284. 0018.0038.1180.09F0.0000.10C4  ;3005  RC[0E]_K[.4]
U 10C4. 0000.003D.0180.0800.0000.1150  ;3006 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 10C5. 0818.0038.7980.0800.0000.10C6  ;3007  D_K[.30]  ; GENERATE THE ADDRESS OF
U 10C6. 0819.0001.0980.0800.0000.1156  ;3008 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; THE FPA UCODE.
U 10C7. 0003.003C.5980.3E80.0000.1285  ;3009  ID[ACC.2]_D,R[0]_0
U 1285. 0058.0038.4580.0A88.0000.1286  ;3010  ALU_K[.8000],R[1]_ALU,RIGHT,SI/ZERO ; EXPONENT = 80
U 1286. 0818.0038.0D80.0800.0000.1287  ;3011  D_K[.3]
U 1287. 0000.00FC.3F80.3C00.0000.1288  ;3012  TRAP,FPA,ID[PSL]_D ; TRAP THE FPA TO 2E. PSL=3.
U 1288. 0000.003C.0180.0A08.0000.1289  ;3013  LAB_R[1]  ; FPA AT 2E
U 1289. 0000.007C.0180.0A08.0000.128A  ;3014  LAB_R[1],CPSYNC
U 128A. 0000.003C.0180.0A08.0000.128B  ;3015  LAB_R[1]  ; FPA AT 6E
U 128B. 0000.003C.0180.0A08.0000.128C  ;3016  LAB_R[1]  ; FPA AT 2C
U 128C. 0000.003C.0180.0A08.0000.128D  ;3017  LAB_R[1]  ; FPA AT 2D
U 128D. 0000.003C.0180.0A08.0000.128E  ;3018  LAB_R[1]  ; FPA AT 2C
U 128E. 0000.003C.0180.0A08.0000.128F  ;3019  LAB_R[1]  ; FPA AT 2D
U 128F. 0000.003C.0180.0A08.0000.1290  ;3020  LAB_R[1]  ; FPA AT 2C
U 1290. 0000.003C.0180.0A08.0000.1291  ;3021  LAB_R[1]  ; FPA AT 2D
```

```
U 1291, 0000.003C.0180.0A08.0000.1292  ;3022  LAB_R[1]  ; FPA AT 2C
U 1292, 0000.003C.0180.0A08.0000.1293  ;3023  LAB_R[1]  ; FPA AT 2D
U 1293, 0000.003C.0180.0A08.0000.1294  ;3024  LAB_R[1]  ; FPA AT 2C
U 1294, 0000.003C.0180.0A00.0000.1295  ;3025  LAB_R[0]  ; AT FPA 2D
U 1295, 0000.003C.0180.0A00.1800.1296  ;3026  LAB_R[0],MSC/LOAD.ACC.CC ; FPA AT 2C
U 1296, 0000.003C.3DF0.2E00.0000.1297  ;3027  LAB_R[0],Q_ID[PSL] ; FPA AT 2D
U 1297, 0C00.003C.3D80.3E00.0000.1298  ;3028  LAB_R[0],ID[PSL]_D,D_Q ; FPA AT 2C
U 1298, 0000.003C.0180.0A00.0000.1299  ;3029  LAB_R[0]  ; FPA AT 2D
U 1299, 0000.003C.0180.0A00.1800.129A  ;3030  LAB_R[0],MSC/LOAD.ACC.CC ; FPA AT 2C
U 129A, 0000.003C.3DF0.2E00.0000.129B  ;3031  LAB_R[0],Q_ID[PSL] ; FPA AT 2D
U 129B, 0819.0034.5D80.09E8.0000.129C  ;3032  ALU_D[AND]K[.7],D_ALU,RC[0D]_ALU
U 129C, 0019.0000.1180.0800.0010.129D  ;3033  ALU_D-K[.4],CLK.UBCC ; CHECK FIRST RESULT.
U 129D, 0000.013C.0180.0800.0000.10C8  ;3034  Z?
U 10C8, 0000.003D.0180.0800.0000.1147  ;3035 =0 ERROR1   ; EALU OUTPUT INCORRECT OR
          ;3036     ; LOAD ACC.CC FAILED.
U 10C9, 0019.2034.5DC0.09E8.0000.129E  ;3037  ALU_Q[AND]K[.7],Q_ALU,RC[0D]_ALU
U 129E, 0019.2000.1180.0800.0010.129F  ;3038  ALU_Q-K[.4],CLK.UBCC ; CHECK THE SECOND RESULT.
U 129F, 0000.013C.0180.0800.0000.10CA  ;3039  Z?
U 10CA, 0000.003D.0180.0800.0000.1147  ;3040 =0 ERROR1   ; PR CONTENTS AND AMX/PR
          ;3041     ; INCORRECT.
U 10CB, 0000.003C.0180.0800.0000.10CC  ;3042  NOP
```

```
                ;3043  .PAGE "TEST 25A PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 4"
                ;3044 ;*************************************************************
                ;3045 ;**
                ;3046 ; PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 4
                ;3047 ;
                ;3048 ; TEST DESCRIPTION
                ;3049 ; THIS TEST FLOATS A 1 ACROSS BITS <7:0> OF THE PATTERN 200
                ;3050 ; AND PUTS EACH PATTERN GENERATED ON THE OUTPUT
                ;3051 ; OF THE EALU TO LOAD THE PSL CONDITION CODES. THIS SHOULD
                ;3052 ; CAUSE THE Z BIT TO SET. THEN EACH PATTERN IS LOADED INTO
                ;3053 ; THE PR REGISTER, THE PR IS GATED THRU THE EALU AND THE
                ;3054 ; PSL CONDITION CODES ARE AGAIN LOADED AND CHECKED.
                ;3055 ;
                ;3056 ; FPA UCODE USED
                ;3057 ; LOCATION CONTENTS
                ;3058 ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3059 ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3060 ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3061 ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3062 ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3063 ; 6F:  BEN/7,NAD/0
                ;3064 ;
                ;3065 ; LOGIC DESCRIPTION
                ;3066 ;
                ;3067 ; ERROR DESCRIPTION
                ;3068 ; EXPECTED PSL CONDITION CODES, Z V C BITS <2:0>
                ;3069 ; GOT PSL CONDITION CODES, Z V C BITS <2:0>
                ;3070 ; EXPECTED EALU OUTPUT <09:00>
                ;3071 ;
                ;3072 ; SYNC POINT DESCRIPTION
                ;3073 ;
                ;3074 ;--
                ;3075 ;*************************************************************
                ;3076 =0
U 10CC, 0000.003D.0180.0800.0000.1134  ;3077 CCT4: NEWTST
U 10CD, 0018.0038.0D80.09F8.0000.12A0  ;3078  RC[0F]_K[.3]
U 12A0, 0018.0038.1180.09F0.0000.12A1  ;3079  RC[0E]_K[.4]
U 12A1, 0018.0038.5D80.0AA8.0000.12A2  ;3080  R[5]_K[.7]  ; INITIALIZE A COUNTER.
U 12A2, 0000.003C.D980.0800.0084.72A3  ;3081  SC_K[.9]
U 12A3, 0003.001C.0180.0AA0.0000.10CE  ;3082  ALU_NOT.MASK,R[4]_ALU ; R[4]=200
U 10CE, 0000.003D.0180.0800.0000.1146  ;3083 =0 ERLOOP
U 10CF, 0000.003C.0180.0800.0000.10D0  ;3084  NOP
                ;3085 =0
U 10D0, 0000.003D.0180.0800.0000.1150  ;3086 CCT4A: CALL,J/FPINIT  ; INITIALIZE THE FPA
U 10D1, 0800.003C.0180.0A20.0000.12A4  ;3087  D_R[4]   ; GENERATE THE TEST DATA PATTERN.
U 12A4, 0000.003C.0180.0A28.0082.12A5  ;3088  SC_R[5]
U 12A5, 0001.001C.5D80.09E0.0084.92A6  ;3089  ALU_D.ORNOT.MASK,RC[0C]_ALU,SC_SC+K[.7]
U 12A6, 0003.001C.0180.0A90.0000.12A7  ;3090  ALU_NOT.MASK,R[2]_ALU
U 12A7, 0003.003C.0180.0A80.0000.12A8  ;3091  R[0]_0
U 12A8, 0058.0038.4580.0A88.0000.12A9  ;3092  ALU_K[.8000],R[1]_ALU.RIGHT,SI/ZERO ; EXPONENT=80
U 12A9, 0818.0038.7980.0800.0000.10D2  ;3093  D_K[.30]  ; GENERATE THE ADDRESS OF THE
U 10D2, 0819.0001.0980.0800.0000.1156  ;3094 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 10D3, 0000.003C.5980.3C00.0000.12AB  ;3095  ID[ACC.2]_D
U 12AB, 0818.0038.0D80.0800.0000.12AC  ;3096  D_K[.3]
U 12AC, 0000.00FC.3F80.3C00.0000.12AD  ;3097  TRAP.FPA,ID[PSL]_D ; TRAP THE FPA TO 2E, PSL=3.
```

```
U 12AD. 0000,003C,0180,0A08,0000,12AE  ;3098  LAB_R[1]  ; AT FPA 2E
U 12AE. 0000,007C,0180,0A08,0000,12AF  ;3099  LAB_R[1],CPSYNC
U 12AF. 0000,003C,0180,0A08,0000,12B0  ;3100  LAB_R[1]  ; AT FPA 6E
U 12B0. 0000,003C,0180,0A08,0000,12B1  ;3101  LAB_R[1]  ; AT FPA 2C
U 12B1. 0000,003C,0180,0A08,0000,12B2  ;3102  LAB_R[1]  ; AT FPA 2D
U 12B2. 0000,003C,0180,0A08,0000,12B3  ;3103  LAB_R[1]  ; AT FPA 2C
U 12B3. 0000,003C,0180,0A08,0000,12B4  ;3104  LAB_R[1]  ; AT FPA 2D
U 12B4. 0000,003C,0180,0A08,0000,12B5  ;3105  LAB_R[1]  ; AT FPA 2C
U 12B5. 0000,003C,0180,0A10,0000,12B6  ;3106  LAB_R[2]  ; AT FPA 2D
U 12B6. 0000,003C,0180,0A10,0000,12B7  ;3107  LAB_R[2]  ; AT FPA 2C
U 12B7. 0000,003C,0180,0A00,0000,12B8  ;3108  LAB_R[0]  ; AT FPA 2D
U 12B8. 0000,003C,0180,0A00,1800,12B9  ;3109  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 12B9. 0000,003C,3DF0,2E00,0000,12BA  ;3110  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 12BA. 0C00,003C,3D80,3E00,0000,12BB  ;3111  LAB_R[0],ID[PSL]_D,D_Q ; AT FPA 2C
U 12BB. 0000,003C,0180,0A00,0000,12BC  ;3112  LAB_R[0]  ; AT FPA 2D
U 12BC. 0000,003C,0180,0A00,1800,12BD  ;3113  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 12BD. 0000,003C,3DF0,2E00,0000,12BE  ;3114  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 12BE. 0819,0034,5D80,09E8,0000,12BF  ;3115  ALU_D[AND]K[.7],D_ALU,RC[0D]_ALU
U 12BF. 0019,0000,1180,0800,0010,12C0  ;3116  ALU_D-K[.4],CLK.UBCC ; CHECK THE FIRST RESULT
U 12C0. 0000,013C,0180,0800,0000,10D4  ;3117  Z?
U 10D4. 0000,003D,0180,0800,0000,1147  ;3118 =0 ERROR1   ; EALU OUTPUT INCORRECT OR
        ;3119     ; LOAD ACC.CC FAILED.
U 10D5. 0019,2034,5DC0,09E8,0000,12C1  ;3120  ALU_Q[AND]K[.7],Q_ALU,RC[0D]_ALU
U 12C1. 0019,2000,1180,0800,0010,12C2  ;3121  ALU_Q-K[.4],CLK.UBCC ; CHECK THE SECOND RESULT.
U 12C2. 0000,013C,0180,0800,0000,10D6  ;3122  Z?
U 10D6. 0000,003D,0180,0800,0000,1147  ;3123 =0 ERROR1   ; PR CONTENTS OR AMX/PR FAILED.
U 10D7. 0000,003C,0180,0A28,0010,12C3  ;3124  ALU_R[5],CLK.UBCC ; TEST DONE?
U 12C3. 0000,013C,0180,0800,0000,10D8  ;3125  Z?
U 10D8. 0018,0000,0580,0AA8,0000,10D0  ;3126 =0 R[5]_LA-K[.1],J/CCT4A ; NO SO CONTINUE.
U 10D9. 0000,003C,0180,0800,0000,10DA  ;3127  NOP
```

```
                ;3128  .PAGE "TEST 25B PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 5"
                ;3129  ;*****************************************************************
                ;3130  ;**
                ;3131  ; PR <09:08>, EALU <09:08> AND LOAD ACC.CC TEST 5
                ;3132  ;
                ;3133  ; TEST DESCRIPTION
                ;3134  ; THIS TEST GENERATES PATTERNS BY FLOATING A 1 ACROSS BITS
                ;3135  ; <7:0> OF THE EXPONENT WITH BITS <9:8>=0. THESE PATTERNS ARE
                ;3136  ; PLACED ON THE EALU OUTPUT AND THE PSL CONDITION CODES ARE
                ;3137  ; LOADED. THIS SHOULD CAUSE THE CONDITION CODES TO CLEAR. THEN THE
                ;3138  ; PATTERN IS LOADED INTO THE PR REGISTER. THE PR IS GATED
                ;3139  ; THRU THE EALU AND THE CONDITION CODES ARE LOADED AND CHECKED
                ;3140  ; AGAIN.
                ;3141  ;
                ;3142  ; FPA UCODE USED
                ;3143  ; LOCATION CONTENTS
                ;3144  ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3145  ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3146  ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3147  ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3148  ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3149  ; 6F:  BEN/7,NAD/0
                ;3150  ;
                ;3151  ; LOGIC DESCRIPTION
                ;3152  ;
                ;3153  ; ERROR DESCRIPTION
                ;3154  ; EXPECTED PSL CONDITION CODES, Z V C BITS <2:0>
                ;3155  ; GOT PSL CONDITION CODES, Z V C BITS <2:0>
                ;3156  ; EXPECTED EALU OUTPUT <09:00>
                ;3157  ;
                ;3158  ; SYNC POINT DESCRIPTION
                ;3159  ;
                ;3160  ;--
                ;3161  ;*****************************************************************
                ;3162  =0
U 10DA. 0000.003D.0180.0800.0000.1134   ;3163 CCT5: NEWTST
U 10DB. 001B.0038.0D80.09F8.0000.12C4   ;3164  RC[0F]_K[.3]
U 12C4. 0003.003C.0180.09F0.0000.12C5   ;3165  RC[0E]_0
U 12C5. 001B.0038.5D80.0AA8.0000.10DC   ;3166  R[5]_K[.7]  ; INITIALIZE A COUNTER
U 10DC. 0000.003D.0180.0800.0000.1146   ;3167 =0 ERLOOP
U 10DD. 0000.003C.0180.0800.0000.10DE   ;3168  NOP
                ;3169  =0
U 10DE. 0000.003D.0180.0800.0000.1150   ;3170 CCT5A: CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 10DF. 0000.003C.0180.0A28.0082.12C6   ;3171  SC_R[5]   ; GENERATE THE TEST DATA PATTERN.
U 12C6. 0003.001C.5D80.09E0.0084.92C7   ;3172  ALU_NOT.MASK,RC[0C]_ALU,SC_SC+K[.7]
U 12C7. 0003.001C.0180.0A90.0000.12C8   ;3173  ALU_NOT.MASK,R[2]_ALU
U 12C8. 0003.003C.0180.0A80.0000.12C9   ;3174  R[0]_0
U 12C9. 0058.0038.4580.0A88.0000.12CA   ;3175  ALU_K[.8000],R[1]_ALU.RIGHT,SI/ZERO ; EXPONENT = 80
U 12CA. 0818.0038.7980.0800.0000.10E0   ;3176  D_K[.30]  ; GENERATE THE ADDRESS OF
U 10E0. 0819.0001.0980.0800.0000.1156   ;3177 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; THE FPA UCODE.
U 10E1. 0000.003C.5980.3C00.0000.12CB   ;3178  ID[ACC.2]_D
U 12CB. 0818.0038.5D80.0800.0000.12CC   ;3179  D_K[.7]
U 12CC. 0000.00FC.3F80.3C00.0000.12CD   ;3180  TRAP,FPA,ID[PSL]_D ; TRAP TO FPA 2E, PSL=7.
U 12CD. 0000.003C.0180.0A08.0000.12CE   ;3181  LAB_R[1]  ; AT FPA 2E
U 12CE. 0000.007C.0180.0A08.0000.12CF   ;3182  LAB_R[1],CPSYNC
```

```
U 12CF. 0000.003C.0180.0A08.0000.12D0  ;3183  LAB_R[1]  ; AT FPA 6E
U 12D0. 0000.003C.0180.0A08.0000.12D1  ;3184  LAB_R[1]  ; AT FPA 2C
U 12D1. 0000.003C.0180.0A08.0000.12D2  ;3185  LAB_R[1]  ; AT FPA 2D
U 12D2. 0000.003C.0180.0A08.0000.12D3  ;3186  LAB_R[1]  ; AT FPA 2C
U 12D3. 0000.003C.0180.0A08.0000.12D4  ;3187  LAB_R[1]  ; AT FPA 2D
U 12D4. 0000.003C.0180.0A08.0000.12D5  ;3188  LAB_R[1]  ; AT FPA 2C
U 12D5. 0000.003C.0180.0A08.0000.12D6  ;3189  LAB_R[1]  ; AT FPA 2D
U 12D6. 0000.003C.0180.0A08.0000.12D7  ;3190  LAB_R[1]  ; AT FPA 2C
U 12D7. 0000.003C.0180.0A08.0000.12D8  ;3191  LAB_R[1]  ; AT FPA 2D
U 12D8. 0000.003C.0180.0A08.0000.12D9  ;3192  LAB_R[1]  ; AT FPA 2C
U 12D9. 0000.003C.0180.0A08.0000.12DA  ;3193  LAB_R[1]  ; AT FPA 2D
U 12DA. 0000.003C.0180.0A08.0000.12DB  ;3194  LAB_R[1]  ; AT FPA 2C
U 12DB. 0000.003C.0180.0A08.0000.12DC  ;3195  LAB_R[1]  ; AT FPA 2D
U 12DC. 0000.003C.0180.0A08.0000.12DD  ;3196  LAB_R[1]  ; AT FPA 2C
U 12DD. 0000.003C.0180.0A10.0000.12DE  ;3197  LAB_R[2]  ; AT FPA 2D
U 12DE. 0000.003C.0180.0A10.0000.12DF  ;3198  LAB_R[2]  ; AT FPA 2C
U 12DF. 0000.003C.0180.0A00.0000.12E0  ;3199  LAB_R[0]  ; AT FPA 2D
U 12E0. 0000.003C.0180.0A00.1800.12E1  ;3200  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 12E1. 0000.003C.3DF0.2E00.0000.12E2  ;3201  LAB_R[0],Q_ID[PSL] ; AT FPA 2D
U 12E2. 0C00.003C.3D80.3E00.0000.12E3  ;3202  LAB_R[0],ID[PSL]_D.D_Q ; AT FPA 2C
U 12E3. 0000.003C.0180.0A00.0000.12E4  ;3203  LAB_R[0]  ; AT FPA 2D
U 12E4. 0000.003C.0180.0A00.1800.12E5  ;3204  LAB_R[0],MSC/LOAD.ACC.CC ; AT FPA 2C
U 12E5. 0000.003C.3DF0.2E00.0000.12E6  ;3205  LAB_R[0],Q_ID[PSL]
U 12E6. 0819.0034.5D80.09E8.0010.12E7  ;3206  ALU_D[AND]K[.7],D_ALU,RC[0D]_ALU,CLK.UBCC ; CHECK 1ST RESULT
U 12E7. 0000.013C.0180.0800.0000.10E2  ;3207  Z?
U 10E2. 0000.003D.0180.0800.0000.1147  ;3208 =0 ERROR1   ; EALU OUTPUT, PR REGISTER OR
        ;3209     ; LOAD ACC.CC FAILED.
U 10E3. 0019.2034.5DC0.09E8.0010.12E8  ;3210  ALU_Q[AND]K[.7],Q_ALU,RC[0D]_ALU,CLK.UBCC ; CHECK 2ND RESULT
U 12E8. 0000.013C.0180.0800.0000.10E4  ;3211  Z?
U 10E4. 0000.003D.0180.0800.0000.1147  ;3212 =0 ERROR1   ; EALU OUTPUT, PR REGISTER OR
        ;3213     ; LOAD ACC.CC FAILED.
U 10E5. 0000.003C.0180.0A28.0010.12E9  ;3214  ALU_R[5],CLK.UBCC ; FINISHED TEST?
U 12E9. 0000.013C.0180.0800.0000.10E6  ;3215  Z?
U 10E6. 0018.0000.0580.0AA8.0000.10DE  ;3216 =0 R[5]_LA-K[.1],J/CCT5A ; NO SO CONTINUE.
U 10E7. 0000.003C.0180.0800.0000.10E8  ;3217  NOP
```

```
                ;3218  .PAGE "TEST 25C BEN/7 TEST 1"
                ;3219  ;*****************************************************************
                ;3220  ;**
                ;3221  ; BEN/7 TEST 1
                ;3222  ;
                ;3223  ; TEST DESCRIPTION
                ;3224  ; THIS IS A TEST OF BEN/7 WITH PR=000.
                ;3225  ;
                ;3226  ; FPA UCODE USED
                ;3227  ; LOCATION CONTENTS
                ;3228  ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3229  ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3230  ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3231  ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3232  ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3233  ; 6F:  BEN/7,NAD/0
                ;3234  ;
                ;3235  ; LOGIC DESCRIPTION
                ;3236  ;
                ;3237  ; ERROR DESCRIPTION
                ;3238  ; EXPECTED NEXT UADDRESS, 0
                ;3239  ; GOT NEXT UADDRESS
                ;3240  ; CONTENTS OF PR WHEN BEN/7 EXECUTED
                ;3241  ;
                ;3242  ; SYNC POINT DESCRIPTION
                ;3243  ;
                ;3244  ;--
                ;3245  ;*****************************************************************
                ;3246  =0
U 10E8. 0000.003D.0180.0800.0000.1134   ;3247 B7T1: NEWTST
U 10E9. 001B.0038.0D80.09F8.0000.12EA   ;3248  RC[0F]_K[.3]
U 12EA. 0003.003C.0180.09F0.0000.12EB   ;3249  RC[0E]_0
U 12EB. 0003.003C.0180.09E0.0000.10EA   ;3250  RC[0C]_0
U 10EA. 000D.003D.0180.0800.0000.1150   ;3251 =0 CALL,J/FPINIT   ; INITIALIZE THE FPA.
U 10EB. 081B.0038.7980.0800.0000.10EC   ;3252  D_K[.30]   ; GENERATE THE ADDRESS OF THE
U 10EC. 0819.0001.0980.0800.0000.1156   ;3253 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE
U 10ED. 0003.003C.5980.3E80.0000.12EC   ;3254  ID[ACC.2]_D,R[0]_0
U 12EC. 000D.00FC.0380.0800.0000.12ED   ;3255  TRAP.FPA   ; TRAP TO FPA 2E
U 12ED. 000D.003C.0180.0A00.0000.12EE   ;3256  LAB_R[0]   ; AT FPA 2E
U 12EE. 000D.007C.0180.0A00.0000.12EF   ;3257  LAB_R[0],CPSYNC
U 12EF. 000D.003C.0180.0A00.0000.12F0   ;3258  LAB_R[0]   ; AT FPA 6E
U 12F0. 000D.007C.0180.0A00.0000.12F1   ;3259  LAB_R[0],CPSYNC   ; AT FPA 2C
U 12F1. 000D.003C.0180.0A00.0000.12F2   ;3260  LAB_R[0]   ; AT FPA 2F
U 12F2. 000D.003C.59F0.2C00.0000.12F3   ;3261  Q_ID[ACC.2]   ; AT 6F, GET NAD
U 12F3. 0019.2034.8180.09E8.0010.12F4   ;3262  ALU_Q[AND]K[.3FF],RC[0D]_ALU,CLK.UBCC ; CHECK THE RESULT
U 12F4. 000D.013C.0180.0800.0000.10EE   ;3263  Z?
U 10EE. 000D.003D.0180.0800.0000.1147   ;3264 =0 ERROR1    ; NAD INCORRECT
U 10EF. 000D.003C.0180.0800.0000.10F0   ;3265  NOP
```

```
                ;3266  .PAGE "TEST 25D BEN/7 TEST 2"
                ;3267  ;*****************************************************************
                ;3268  ;**
                ;3269  ; BEN/7 TEST 2
                ;3270  ;
                ;3271  ; TEST DESCRIPTION
                ;3272  ; THIS IS A TEST OF BEN/7 WITH PR=0XX, PR <9:8>=0 AND
                ;3273  ; PR <7:0> IS A PATTERN GENERATED BY FLOATING A 1 THRU 0'S.
                ;3274  ;
                ;3275  ; FPA UCODE USED
                ;3276  ; LOCATION CONTENTS
                ;3277  ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3278  ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3279  ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3280  ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3281  ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3282  ; 6F:  BEN/7,NAD/0
                ;3283  ;
                ;3284  ; LOGIC DESCRIPTION
                ;3285  ;
                ;3286  ; ERROR DESCRIPTION
                ;3287  ; EXPECTED NEXT UADDRESS, 2
                ;3288  ; GOT NEXT UADDRESS
                ;3289  ; CONTENTS OF PR WHEN BEN/7 EXECUTED
                ;3290  ;
                ;3291  ; SYNC POINT DESCRIPTION
                ;3292  ;
                ;3293  ;--
                ;3294  ;*****************************************************************
                ;3295  =0
U 10F0, 0000.003D.0180.0800.0000.1134  ;3296 B7T2: NEWTST
U 10F1, 0018.0038.0D80.09F8.0000.12F5  ;3297  RC[0F]_K[.3]
U 12F5, 0018.0038.0980.09F0.0000.12F6  ;3298  RC[0E]_K[.2]
U 12F6, 0003.003C.0180.09E0.0000.10F2  ;3299  RC[0C]_0
U 10F2, 0000.003D.0180.0800.0000.1146  ;3300 =0 ERLOOP
U 10F3, 0000.003C.0180.0800.0000.10F4  ;3301  NOP
                ;3302  =0
U 10F4, 0000.003D.0180.0800.0000.1150  ;3303 B7T2A: CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 10F5, 0000.003C.0180.0A28.0082.12F7  ;3304  SC_R[5]   ; GENERATE THE TEST DATA PATTERN.
U 12F7, 0003.001C.5D80.09E0.0084.92F8  ;3305  ALU_NOT.MASK,RC[0C]_ALU,SC_SC+K[.7]
U 12F8, 0003.001C.0180.0A90.0000.12F9  ;3306  ALU_NOT.MASK,R[2]_ALU
U 12F9, 0818.0038.7980.0800.0000.10F6  ;3307  D_K[.30]   ; GENERATE THE ADDRESS OF THE
U 10F6, 0819.0001.0980.0800.0000.1156  ;3308 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE
U 10F7, 0003.003C.5980.3E80.0000.12FA  ;3309  ID[ACC.2]_D,R[0]_0
U 12FA, 0000.00FC.0380.0800.0000.12FB  ;3310  TRAP,FPA   ; TRAP TO FPA 2E
U 12FB, 0000.003C.0180.0A00.0000.12FC  ;3311  LAB_R[0]   ; AT FPA 2E
U 12FC, 0000.007C.0180.0A00.0000.12FD  ;3312  LAB_R[0],CPSYNC
U 12FD, 0000.003C.0180.0A10.0000.12FE  ;3313  LAB_R[2]   ; AT FPA 6E
U 12FE, 0000.003C.0180.0A10.0000.12FF  ;3314  LAB_R[2]   ; AT FPA 2C
U 12FF, 0000.003C.0180.0A00.0000.1300  ;3315  LAB_R[0]   ; AT FPA 2D
U 1300, 0000.007C.0180.0A00.0000.1301  ;3316  LAB_R[0],CPSYNC   ; AT FPA 2C
U 1301, 0000.003C.0180.0A00.0000.1302  ;3317  LAB_R[0]   ; AT FPA 2F
U 1302, 0000.003C.59F0.2C00.0000.1303  ;3318  Q_ID[ACC.2]   ; AT 6F, GET NAD
U 1303, 0019.2034.81C0.09E8.0000.1304  ;3319  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU ; CHECK THE RESULT
U 1304, 0019.2000.0980.0800.0010.1305  ;3320  ALU_Q-K[.2],CLK.UBCC
```

```
U 1305, 0000,013C,0180,0800,0000,10F8  ;3321  Z?
U 10F8, 0000,003D,0180,0800,0000,1147  ;3322 =0 ERROR1    ; NAD INCORRECT
U 10F9, 0000,003C,0180,0A28,0010,1306  ;3323  ALU_R[5],CLK.UBCC
U 1306, 0000,013C,0180,0800,0000,10FA  ;3324  Z?
U 10FA, 0018,0000,0580,0AA8,0000,10F4  ;3325 =0 R[5]_LA-K[.1],J/B7T2A ; CONTINUE TESTING
U 10FB, 0000,003C,0180,0800,0000,10FC  ;3326  NOP
```

```
                ;3327  .PAGE "TEST 25E BEN/7 TEST 3"
                ;3328  ;*****************************************************************
                ;3329  ;**
                ;3330  ; BEN/7 TEST 3
                ;3331  ;
                ;3332  ; TEST DESCRIPTION
                ;3333  ; THIS IS A TEST OF BEN/7 WITH PR=100.
                ;3334  ;
                ;3335  ; FPA UCODE USED
                ;3336  ; LOCATION CONTENTS
                ;3337  ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3338  ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3339  ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3340  ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3341  ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3342  ; 6F:  BEN/7,NAD/0
                ;3343  ;
                ;3344  ; LOGIC DESCRIPTION
                ;3345  ;
                ;3346  ; ERROR DESCRIPTION
                ;3347  ; EXPECTED NEXT UADDRESS, 3
                ;3348  ; GOT NEXT UADDRESS
                ;3349  ; CONTENTS OF PR WHEN BEN/7 EXECUTED
                ;3350  ;
                ;3351  ; SYNC POINT DESCRIPTION
                ;3352  ;
                ;3353  ;--
                ;3354  ;*****************************************************************
                ;3355  =0
U 10FC. 0000.003D.0180.0800.0000.1134   ;3356 B7T3: NEWTST
U 10FD. 0018.0038.0D80.09F8.0000.1307   ;3357  RC[0F]_K[.3]
U 1307. 0018.0038.0D80.09F0.0000.1308   ;3358  RC[0E]_K[.3]
U 1308. 0000.003C.2180.0800.0084.7309   ;3359  SC_K[8]
U 1309. 0003.001C.0180.09E0.0000.10FE   ;3360  ALU_NOT.MASK,RC[0C]_ALU
U 10FE. 0000.003D.0180.0800.0000.1150   ;3361 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 10FF. 0818.0038.7980.0800.0000.110A   ;3362  D_K[.30]  ; GENERATE THE ADDRESS OF THE
U 110A. 0819.0001.0980.0800.0000.1156   ;3363 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE
U 110B. 0003.003C.5980.3E80.0000.130A   ;3364  ID[ACC.2]_D,R[0]_0
U 130A. 0058.0038.4580.0A88.0000.130B   ;3365  ALU_K[.8000],R[1]_ALU.RIGHT.SI/ZERO ; EXPONENT = 80
U 130B. 0000.00FC.0380.0800.0000.130C   ;3366  TRAP.FPA  ; TRAP TO FPA 2E
U 130C. 0000.003C.0180.0A08.0000.130D   ;3367  LAB_R[1]  ; AT FPA 2E
U 130D. 0000.007C.0180.0A08.0000.130E   ;3368  LAB_R[1],CPSYNC
U 130E. 0000.003C.0180.0A08.0000.130F   ;3369  LAB_R[1]  ; AT FPA 6E
U 130F. 0000.003C.0180.0A08.0000.1310   ;3370  LAB_R[1]  ; AT FPA 2C
U 1310. 0000.003C.0180.0A00.0000.1311   ;3371  LAB_R[0]  ; AT FPA 2D
U 1311. 0000.007C.0180.0A00.0000.1312   ;3372  LAB_R[0],CPSYNC  ; AT FPA 2C
U 1312. 0000.003C.0180.0A00.0000.1313   ;3373  LAB_R[0]  ; AT FPA 2F
U 1313. 0000.003C.59F0.2C00.0000.1314   ;3374  Q_ID[ACC.2]  ; AT 6F, GET NAD
U 1314. 0019.2034.81C0.09E8.0000.1315   ;3375  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU ; CHECK THE RESULT
U 1315. 0019.2000.0D80.0800.0010.1316   ;3376  ALU_Q-K[.3],CLK.UBCC
U 1316. 0000.013C.0180.0800.0000.1110   ;3377  Z?
U 1110. 0000.003D.0180.0800.0000.1147   ;3378 =0 ERROR1   ; NAD INCORRECT
U 1111. 0000.003C.0180.0800.0000.1112   ;3379  NOP
```

```
          ;3380  .PAGE "TEST 25F BEN/7 TEST 4"
          ;3381 ;*****************************************************************
          ;3382 ;**
          ;3383 ; BEN/7 TEST 4
          ;3384 ;
          ;3385 ; TEST DESCRIPTION
          ;3386 ; THIS IS A TEST OF BEN/7 WITH PR=300.
          ;3387 ;
          ;3388 ; FPA UCODE USED
          ;3389 ; LOCATION CONTENTS
          ;3390 ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
          ;3391 ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
          ;3392 ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
          ;3393 ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
          ;3394 ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
          ;3395 ; 6F:  BEN/7,NAD/0
          ;3396 ;
          ;3397 ; LOGIC DESCRIPTION
          ;3398 ;
          ;3399 ; ERROR DESCRIPTION
          ;3400 ; EXPECTED NEXT UADDRESS, 1
          ;3401 ; GOT NEXT UADDRESS
          ;3402 ; CONTENTS OF PR WHEN BEN/7 EXECUTED
          ;3403 ;
          ;3404 ; SYNC POINT DESCRIPTION
          ;3405 ;
          ;3406 ;--
          ;3407 ;*****************************************************************
          ;3408 =0
U 1112, 0000,003D,0180,0800,0000,1134  ;3409 B7T4: NEWTST
U 1113, 0018,0038,0D80,09F8,0000,1317  ;3410  RC[0F]_K[.3]
U 1317, 0018,0038,0580,09F0,0000,1318  ;3411  RC[0E]_K[.1]
U 1318, 0000,003C,0180,0800,0084,7319  ;3412  SC_K[.8]
U 1319, 0003,001C,0180,09E0,0000,1114  ;3413  ALU_NOT.MASK,RC[0C]_ALU
U 1114, 0000,003D,0180,0800,0000,1150  ;3414 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1115, 0818,0038,7980,0800,0000,1116  ;3415  D_K[.30]  ; GENERATE THE ADDRESS OF THE
U 1116, 0819,0001,0980,0800,0000,1156  ;3416 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE
U 1117, 0003,003C,5980,3E80,0000,131A  ;3417  ID[ACC.2]_D,R[0]_0
U 131A, 0058,0038,4580,0A88,0000,131B  ;3418  ALU_K[.8000],R[1]_ALU.RIGHT,SI/ZERO ; EXPONENT = 80
U 131B, 0000,00FC,0380,0800,0000,131C  ;3419  TRAP.FPA  ; TRAP TO FPA 2E
U 131C, 0000,003C,0180,0A08,0000,131D  ;3420  LAB_R[1]  ; AT FPA 2E
U 131D, 0000,007C,0180,0A08,0000,131E  ;3421  LAB_R[1].CPSYNC
U 131E, 0000,003C,0180,0A08,0000,131F  ;3422  LAB_R[1]  ; AT FPA 6E
U 131F, 0000,003C,0180,0A08,0000,1320  ;3423  LAB_R[1]  ; AT FPA 2C
U 1320, 0000,003C,0180,0A08,0000,1321  ;3424  LAB_R[1]  ; AT FPA 2D
U 1321, 0000,003C,0180,0A08,0000,1322  ;3425  LAB_R[1]  ; AT FPA 2C
U 1322, 0000,003C,0180,0A08,0000,1323  ;3426  LAB_R[1]  ; AT FPA 2D
U 1323, 0000,003C,0180,0A08,0000,1324  ;3427  LAB_R[1]  ; AT FPA 2C
U 1324, 0000,003C,0180,0A08,0000,1325  ;3428  LAB_R[1]  ; AT FPA 2D
U 1325, 0000,003C,0180,0A08,0000,1326  ;3429  LAB_R[1]  ; AT FPA 2C
U 1326, 0000,003C,0180,0A08,0000,1327  ;3430  LAB_R[1]  ; AT FPA 2D
U 1327, 0000,003C,0180,0A08,0000,1328  ;3431  LAB_R[1]  ; AT FPA 2C
U 1328, 0000,003C,0180,0A00,0000,1329  ;3432  LAB_R[0]  ; AT FPA 2D
U 1329, 0000,007C,0180,0A00,0000,132A  ;3433  LAB_R[0].CPSYNC  ; AT FPA 2C
U 132A, 0000,003C,0180,0A00,0000,132B  ;3434  LAB_R[0]  ; AT FPA 2F
```

```
U 132B, 0000.003C.59F0.2C00.0000.132C  ;3435  Q_ID[ACC.2]  ; AT 6F, GET NAD
U 132C, 0019.2034.81C0.09E8.0000.132D  ;3436  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU ; CHECK THE RESULT
U 132D, 0019.2000.0580.0800.0010.132E  ;3437  ALU_Q-K[.1],CLK.UBCC
U 132E, 0000.013C.0180.0800.0000.1118  ;3438  Z?
U 1118, 0000.003D.0180.0800.0000.1147  ;3439 =0 ERROR1   ; NAD INCORRECT
U 1119, 0000.003C.0180.0800.0000.111A  ;3440  NOP
```

```
                ;3441  .PAGE "TEST 260 BEN/7 TEST 5"
                ;3442 ;*****************************************************************
                ;3443 ;**
                ;3444 ; BEN/7 TEST 5
                ;3445 ;
                ;3446 ; TEST DESCRIPTION
                ;3447 ; THIS IS A TEST OF BEN/7 WITH PR=2XX, PR 9=1, PR 8=0 AND
                ;3448 ; PR <7:0> = A 1 IN A FIELD OF 0'S.
                ;3449 ;
                ;3450 ; FPA UCODE USED
                ;3451 ; LOCATION CONTENTS
                ;3452 ; 2E:  BSC/R,SCR/CPU,EAC/LDAB,NAD/6E,WAIT
                ;3453 ; 6E:  AMXC/LA,EALUC/A,EAC/LDPR,NAD/2C
                ;3454 ; 2C:  BSC/R,SCR/CPU,EAC/LDAB,BEN/4,NAD/2D
                ;3455 ; 2D:  AMXC/PR,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/2C
                ;3456 ; 2F:  AMXC/PR,EALUC/A,EAC/LDPR,NAD/6F
                ;3457 ; 6F:  BEN/7,NAD/0
                ;3458 ;
                ;3459 ; LOGIC DESCRIPTION
                ;3460 ;
                ;3461 ; ERROR DESCRIPTION
                ;3462 ; EXPECTED NEXT UADDRESS, 0
                ;3463 ; GOT NEXT UADDRESS
                ;3464 ; CONTENTS OF PR WHEN BEN/7 EXECUTED
                ;3465 ;
                ;3466 ; SYNC POINT DESCRIPTION
                ;3467 ;
                ;3468 ;--
                ;3469 ;*****************************************************************
                ;3470 =0
U 111A, 0000.003D.0180.0800,0000.1134  ;3471 B7T5: NEWTST
U 111B, 0018.0038.0DB0.09F8,0000.132F  ;3472  RC[0F]_K[.3]
U 132F, 0003.003C.0180.09F0,0000.1330  ;3473  RC[0E]_0
U 1330, 0000.003C.D980.0800,0084.7331  ;3474  SC_K[.9]
U 1331, 0003.001C.0180.0AB0,0000.1332  ;3475  ALU_NOT.MASK,R[6]_ALU
U 1332, 0018.0038.5DB0.0AA8,0000.111C  ;3476  R[5]_K[.7]
U 111C, 0000.003D.0180.0800,0000.1146  ;3477 =0 ERLOOP
U 111D, 0000.003C.0180.0800,0000.111E  ;3478  NOP
                ;3479 =0
U 111E, 0000.003D.0180.0800,0000.1150  ;3480 B7T5A: CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 111F, 0000.003C.0180.0A28,0082.1333  ;3481  SC_R[5]
U 1333, 0800.003C.0180.0A30,0000.1334  ;3482  D_R[6]
U 1334, 0001.001C.5DB0.09E0,0084.9335  ;3483  ALU_D.ORNOT.MASK,RC[0C]_ALU,SC_SC+K[.7]
U 1335, 0003.001C.0180.0A90,0000.1336  ;3484  ALU_NOT.MASK,R[2]_ALU
U 1336, 0058.0038.4580.0A88,0000.1337  ;3485  ALU_K[.8000],R[1]_ALU.RIGHT,SI/ZERO
U 1337, 0818.0038.7980.0800,0000.1120  ;3486  D_K[.30]  ; GENERATE THE ADDRESS OF THE
U 1120, 0819.0001.0980.0800,0000.1156  ;3487 =0 ALU_D-K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE
U 1121, 0003.003C.5980.3E80,0000.1338  ;3488  ID[ACC.2]_D,R[0]_0
U 1338, 0000.00FC.0380.0800,0000.1339  ;3489  TRAP.FPA  ; TRAP TO FPA 2E
U 1339, 0000.003C.0180.0A08,0000.133A  ;3490  LAB_R[1]  ; AT FPA 2E
U 133A, 0000.007C.0180.0A08,0000.133B  ;3491  LAB_R[1].CPSYNC
U 133B, 0000.003C.0180.0A08,0000.133C  ;3492  LAB_R[1]  ; AT FPA 6E
U 133C, 0000.003C.0180.0A08,0000.133D  ;3493  LAB_R[1]  ; AT FPA 2C
U 133D, 0000.003C.0180.0A08,0000.133E  ;3494  LAB_R[1]  ; AT FPA 2D
U 133E, 0000.003C.0180.0A08,0000.133F  ;3495  LAB_R[1]  ; AT FPA 2C
```

```
U 133F. 0000.003C.0180.0A08.0000.1340  ;3496  LAB_R[1]  ; AT FPA 2D
U 1340. 0000.003C.0180.0A08.0000.1341  ;3497  LAB_R[1]  ; AT FPA 2C
U 1341. 0000.003C.0180.0A10.0000.1342  ;3498  LAB_R[2]  ; AT FPA 2D
U 1342. 0000.003C.0180.0A10.0000.1343  ;3499  LAB_R[2]  ; AT FPA 2C
U 1343. 0000.003C.0180.0A00.0000.1344  ;3500  LAB_R[0]  ; AT FPA 2D
U 1344. 0000.007C.0180.0A00.0000.1345  ;3501  LAB_R[0],CPSYNC  ; AT FPA 2C
U 1345. 0000.003C.0180.0A00.0000.1346  ;3502  LAB_R[0]  ; AT FPA 2F
U 1346. 0000.003C.59F0.2C00.0000.1347  ;3503  Q_ID[ACC.2]  ; AT 6F, GET NAD
U 1347. 0019.2034.8180.09E8.0010.1348  ;3504  ALU_Q[AND]K[.3FF],RC[0D]_ALU,CLK.UBCC ; CHECK THE RESULT
U 1348. 0000.013C.0180.0800.0000.1122  ;3505  Z?
U 1122. 0000.003D.0180.0800.0000.1147  ;3506 =0 ERROR1  ; NAD INCORRECT
U 1123. 0000.003C.0180.0A28.0010.1349  ;3507  ALU_R[5],CLK.UBCC
U 1349. 0000.013C.0180.0800.0000.1124  ;3508  Z?
U 1124. 0018.0000.0580.0AA8.0000.111E  ;3509 =0 R[5]_LA-K[.1],J/B7T5A
U 1125. 0000.003C.0180.0800.0000.1126  ;3510  NOP
        ;3511
```

```
            ;3512  .PAGE "TEST 261 RESERVED"
            ;3513 =0
U 1126, 0000,003D,0180,0800,0000,1134  ;3514 D1: NEWTST
U 1127, 0000,003C,0180,0800,0000,1128  ;3515  NOP
            ;3516
```

```
            ;3517  .PAGE "TEST 262 RESERVED"
            ;3518 =0
U 1128. 0000,003D,0180,0800,0000,1134  ;3519 D2: NEWTST
U 1129. 0000,003C,0180,0800,0000,112A  ;3520  NOP
            ;3521
```

```
            ;3522  .PAGE "DUMMY NEWTST"
            ;3523 =0
U 112A. 0000.003D.0180.0800.0000.1134  ;3524 D3: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2494
ESKARSD.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29          Page  10
;                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 1918= 1919=
U 1008  1995= 1996= 1998= 1999= 2000= 2001= 2003= 2004=
U 1010  2023= 2024= 2070= 2071= 2072= 2073= 2074= 2075=
U 1018  2082= 2083= 2084= 2085= 2092= 2093= 2121= 2122=
U 1020  2123= 2124= 2125= 2126= 2133= 2134= 2135= 2136=
U 1028  2143= 2144= 2176= 2177= 2180= 2181= 2182= 2183=
U 1030  2184= 2185= 2192= 2193= 2195= 2196= 2197= 2198=
U 1038  2205= 2206= 2239= 2240= 2246= 2247= 2249= 2250=
U 1040  2251= 2252= 2263= 2264= 2266= 2267= 2294= 2295=
U 1048  2301= 2302= 2304= 2305= 2306= 2307= 1939: 1875
U 1050  2321= 2322= 2324= 2325= 2354= 2355= 2361= 2362=
U 1058  2364= 2365= 1942: 1876 2366= 2367= 1945: 1877
U 1060  2380= 2381= 2383= 2384= 2411= 2412= 2415= 2416=
U 1068  2420= 2421= 2433= 2434= 2463= 2464= 2470= 2471=
U 1070  2473= 2474= 2475= 2476= 2489= 2490= 2492= 2493=
U 1078  2525= 2526= 2532= 2533= 2535= 2536= 2537= 2538=
U 1080  2551= 2552= 2554= 2555= 2588= 2589= 2590= 2591=
U 1088  2596= 2597= 2608= 2609= 2613= 2614= 2641= 2642=
U 1090  2653= 2654= 2655= 2656= 2667= 2668= 2696= 2697=
U 1098  2708= 2709= 2710= 2711= 2722= 2723= 2753= 2754=
U 10A0  2765= 2766= 2767= 2768= 2780= 2781= 2810= 2811=
U 10A8  2822= 2823= 2824= 2825= 2836= 2837= 2871= 2872=
U 10B0  2875= 2876= 2877= 2878= 2893= 2895= 2898= 2900=
U 10B8  2933= 2934= 2938= 2939= 2940= 2941= 2960= 2962=
U 10C0  2965= 2966= 3000= 3001= 3006= 3007= 3008= 3009=
U 10C8  3035= 3037= 3040= 3042= 3077= 3078= 3083= 3084=
U 10D0  3086= 3087= 3094= 3095= 3118= 3120= 3123= 3124=
U 10D8  3126= 3127= 3163= 3164= 3167= 3168= 3170= 3171=
U 10E0  3177= 3178= 3208= 3210= 3212= 3214= 3216= 3217=
U 10E8  3247= 3248= 3251= 3252= 3253= 3254= 3264= 3265=
U 10F0  3296= 3297= 3300= 3301= 3303= 3304= 3308= 3309=
U 10F8  3322= 3323= 3325= 3326= 3356= 3357= 3361= 3362=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1878 3363= 3364= 1870: 1871: 1872: 1873:
U 1110  3378= 3379= 3409= 3410= 3414= 3415= 3416= 3417=
U 1118  3439= 3440= 3471= 3472= 3477= 3478= 3480= 3481=
U 1120  3487= 3488= 3506= 3507= 3509= 3510= 3514= 3515=
U 1128  3519= 3520= 3524= 1879 1880 1900 1901 1902
U 1130  1903 1904 1905 1906 1916 1917 1920 1921
U 1138  1922 1923 1924 1925 1926 1927 1928 1929
U 1140  1930 1931 1932 1933 1934 1935 1936 1937
U 1148  1938 1940 1941 1973 1974 1980 1981 1982
U 1150  1993 1994 1997 2002 2005 1965: 2015 2016
U 1158  2017 2025 2026 2027 2033 2034 2035 2036
U 1160  2037 2038 2039 2040 2041 2042 2076 2077
U 1168  2078 2079 2080 2081 2086 2087 2088 2089
U 1170  2090 2091 2127 2128 2129 2130 2131 2132
U 1178  2137 2138 2139 2140 2141 2142 2178 2179
U 1180  2186 2187 2188 2189 2190 2191 2194 2199
U 1188  2200 2201 2202 2203 2204 2241 2242 2243
U 1190  2244 2245 2253 2254 2255 2256 2257 2258
U 1198  2259 2260 2261 2262 2265 2296 2297 2298
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2299 2300 2308 2309 2310 2311 2312 2313
U 11A8  2314 2315 2316 2317 2318 2319 2320 2323
U 11B0  2356 2357 2358 2359 2360 2368 2369 2370
U 11B8  2371 2372 2373 2374 2375 2376 2377 2378
U 11C0  2379 2382 2413 2414 2417 2418 2419 2422
U 11C8  2423 2424 2425 2426 2427 2428 2429 2430
U 11D0  2431 2432 2465 2466 2467 2468 2469 2477
U 11D8  2478 2479 2480 2481 2482 2483 2484 2485
U 11E0  2486 2487 2488 2491 2527 2528 2529 2530
U 11E8  2531 2539 2540 2541 2542 2543 2544 2545
U 11F0  2546 2547 2548 2549 2550 2553 2592 2593
U 11F8  2594 2595 2598 2599 2600 2601 2602 2603
U 1200  2604 2605 2606 2607 2610 2611 2612 2643
U 1208  2644 2645 2646 2647 2648 2649 2650 2651
U 1210  2652 2657 2658 2659 2660 2661 2662 2663
U 1218  2664 2665 2666 2698 2699 2700 2701 2702
U 1220  2703 2704 2705 2706 2707 2712 2713 2714
U 1228  2715 2716 2717 2718 2719 2720 2721 2755
U 1230  2756 2757 2758 2759 2760 2761 2762 2763
U 1238  2764 2769 2770 2771 2772 2773 2774 2775
U 1240  2776 2777 2778 2779 2812 2813 2814 2815
U 1248  2816 2817 2818 2819 2820 2821 2826 2827
U 1250  2828 2829 2830 2831 2832 2833 2834 2835
U 1258  2873 2874 2879 2880 2881 2882 2883 2884
U 1260  2885 2886 2887 2888 2889 2890 2891 2892
U 1268  2896 2897 2935 2936 2937 2942 2943 2944
U 1270  2945 2946 2947 2948 2949 2950 2951 2952
U 1278  2953 2954 2955 2956 2957 2958 2959 2963
U 1280  2964 3002 3003 3004 3005 3010 3011 3012
U 1288  3013 3014 3015 3016 3017 3018 3019 3020
U 1290  3021 3022 3023 3024 3025 3026 3027 3028
U 1298  3029 3030 3031 3032 3033 3034 3038 3039
U 12A0  3079 3080 3081 3082 3088 3089 3090 3091
U 12A8  3092 3093 1966: 3096 3097 3098 3099 3100
U 12B0  3101 3102 3103 3104 3105 3106 3107 3108
U 12B8  3109 3110 3111 3112 3113 3114 3115 3116
U 12C0  3117 3121 3122 3125 3165 3166 3172 3173
U 12C8  3174 3175 3176 3179 3180 3181 3182 3183
U 12D0  3184 3185 3186 3187 3188 3189 3190 3191
U 12D8  3192 3193 3194 3195 3196 3197 3198 3199
U 12E0  3200 3201 3202 3203 3204 3205 3206 3207
U 12E8  3211 3215 3249 3250 3255 3256 3257 3258
U 12F0  3259 3260 3261 3262 3263 3298 3299 3305
U 12F8  3306 3307 3310 3311 3312 3313 3314 3315
U 1300  3316 3317 3318 3319 3320 3321 3324 3358
U 1308  3359 3360 3365 3366 3367 3368 3369 3370
U 1310  3371 3372 3373 3374 3375 3376 3377 3411
U 1318  3412 3413 3418 3419 3420 3421 3422 3423
U 1320  3424 3425 3426 3427 3428 3429 3430 3431
U 1328  3432 3433 3434 3435 3436 3437 3438 3473
U 1330  3474 3475 3476 3482 3483 3484 3485 3486
U 1338  3489 3490 3491 3492 3493 3494 3495 3496
U 1340  3497 3498 3499 3500 3501 3502 3503 3504
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3505 3508
U 1350 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR5D	858
Used	858
Remaining	

Total microwords used in memory U: 858
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2498
ESKAR5D.MCR                    MICRO2  1P(00)    26-JUL-85  17:01:29     Page  10


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR5D.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
   ESKARSE.MCR   MICRO2 1P(00)   26-JUL-85 17 SEQ 2499

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2230 TEST 263 EALU/A+B TEST
; 2340 TEST 264 EALU/A-B TEST
; 2452 TEST 265 BEN/3 AND BEN/5 BIT 2 TEST
; 2748 TEST 266 ASHF AND EXPONENT DIFFERENCE TEST 1
; 2861 TEST 267 ASHF AND EXPONENT DIFFERENCE TEST 2
; 2966 TEST 268 ASHF AND EXPONENT DIFFERENCE TEST 3
; 3101 TEST 269 ASHF AND EXPONENT DIFFERENCE TEST 4
; 3210 TEST 26A ASHF AND EXPONENT DIFFERENCE TEST 5
; 3315 TEST 26B ASHF AND EXPONENT DIFFERENCE TEST 6
; 3451 TEST 26C RESERVED
; 3456 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR5E
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
                ;1841   .PAGE
                ;1842
                ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
                ;1844 ;*
                ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
                ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
                ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
                ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
                ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
                ;1850 ;
                ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
                ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
                ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
                ;1854 ; R[0F] AND DO A RETURN0.
                ;1855 ;
                ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
                ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
                ;1858 ; TO THE CONSOLE.
                ;1859 ;
                ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003   ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003   ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003   ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003   ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003   ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003   ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003   ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003   ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003   ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003   ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003   ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003   ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003   ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.104A   ;1874 TRPH0: Q_ID[USTACK]
U 104A. 0C00.003C.01E0.0800.0000.106A   ;1875  Q_D,D_Q
U 106A. 0000.003C.8180.3C00.0000.106E   ;1876  ID[USTACK]_D
U 106E. 0C00.003C.01E0.0800.0000.107A   ;1877  Q_D,D_Q
U 107A. 0000.003C.01C0.0A78.0000.107E   ;1878  Q_R[0F]
U 107E. 0001.2024.0180.0800.0010.1109   ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1109. 0000.013C.0180.0800.0000.1002   ;1880  Z?    ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000   ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
                ;1882 ;*
                ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
                ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
                ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
                ;1886 ;
                ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
                ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
                ;1889 ;       FAULT STATUS REGISTER
                ;1890 ;       SBI ERROR REGISTER
                ;1891 ;       TIMEOUT ADDRESS REGISTER
                ;1892 ;       MAINTENANCE REGISTER
                ;1893 ;       CACHE PARITY REGISTER
                ;1894 ;       TB ERROR REGISTER 1
                ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004. 0000.003D.D980.0800.0084.7151  ;1898 =0 SETRCF[.9]
U 1005. 0000.003C.49F0.2C00.0000.1131  ;1899  Q_ID[TBER0]
U 1131. 0001.203C.4DF0.2DB0.0000.1132  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1132. 0001.203C.65F0.2DB8.0000.1133  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 1133. 0001.203C.6DF0.2DD8.0000.1134  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1134. 0001.203C.75F0.2DE0.0000.1135  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 1135. 0001.203C.79F0.2DC8.0000.1136  ;1904  RC[9]_Q,Q_ID[PARITY]
U 1136. 0001.203C.69F0.2DC0.0000.1137  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 1137. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.114B  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 1138. 0000.003C.8580.0800.0084.7139  ;1916 SCOPE: SC_K[.C]
U 1139. 0803.001C.0180.0800.0000.1006  ;1917  ALU_NOT.MASK,D_ALU
U 1006. 0000.003D.7580.3C00.0000.1154  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1007. 0000.003C.65F8.0800.0084.713A  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 113A. 0818.0038.8D80.0800.0000.113B  ;1920  D_K[.1F]    : ...
U 113B. 0D00.003C.0180.0800.0000.113C  ;1921  D_DAL.SC
U 113C. 0000.003C.3D80.3C00.0000.113D  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 113D. 0818.0038.0580.0800.0000.113E  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 113E. 0D00.003C.0180.0800.0000.113F  ;1924  D_DAL.SC
U 113F. 0F00.003C.3180.3C00.0000.1140  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1140. 0000.003C.0180.0800.0000.1141  ;1926  NOP
U 1141. 0000.003C.3980.3C00.0000.1142  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1142. 0000.003C.2980.3C00.0000.1143  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1143. 0000.003C.4980.3C00.0000.1144  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1144. 0818.0038.C180.0800.0000.1145  ;1930  D_K[.FFFF]
U 1145. 0000.003C.6580.3C00.0000.1146  ;1931  ID[SBI.ERR]_D
U 1146. 0F00.003C.6D80.3C00.0000.1147  ;1932  ID[FAULT]_D,D_0
U 1147. 0000.003C.6D80.3C00.0000.1148  ;1933  ID[FAULT]_D
U 1148. 0000.003C.6580.3C00.0000.1149  ;1934  ID[SBI.ERR]_D
U 1149. 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 114A. 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 114B. 0810.0038.0180.0978.0000.114C  ;1937 ERROR1: D_RC[0F]
U 114C. 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 114D. 0810.0038.0180.0978.0000.114E  ;1940 ERROR2: D_RC[0F]
U 114E. 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000,003C,0180,0800,0000,13F3  ;1951 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;1952 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;1953 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;1954 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;1955 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;1956 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;1957 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;1958 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;1959 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;1960 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;1961 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;1962 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;1963 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;1964 13FF: NOP
U 1155, 0001,203E,59F0,2DE8,0000,0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001,203E,59F0,2DE8,0000,0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 114F, 0810,0038,4D80,0978,0000,1150  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1150, 0019,4016,4D80,09F8,0000,0001  ;1974  RC[0F]_D*K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 1151, 0010,0038,01C0,0978,0000,1152  ;1980 SETRCF: Q_RC[0F]
U 1152, 0019,2034,4DC0,0800,0000,1153  ;1981  Q_Q.AND.K[.FF00]
U 1153, 0019,2032,1D80,09F8,0000,0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983 ;
            ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28. WHICH CONTAINS
            ;1986 ;   28: NAD/28   ; NOP
            ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;1990 ;  INITIALIZE ITSELF.
            ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;1992 ;
U 1154, 0818,0038,4580,0800,0000,1156  ;1993 FPINIT: D_K[.8000]
U 1156, 0000,003C,5D80,3C00,0000,1008  ;1994  ID[ACC.CS]_D   ; TURN ON FPA
U 1008, 0818,0039,2D80,0800,0000,115A  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1009, 0000,003C,5980,3C00,0000,1157  ;1996  ID[ACC.2]_D
U 1157, 0000,00FC,0380,0800,0000,100A  ;1997  TRAP.FPA   ; TRAP FPA TO 28.
U 100A, 0018,0039,1980,0800,0200,1160  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 100B, 0F00,003C,0180,0800,0000,100C  ;1999  D_0
U 100C, 0000,003D,0180,0800,0000,115A  ;2000 =0 CALL,J/GENTRA
U 100D, 0000,003C,5980,3C00,0000,1158  ;2001  ID[ACC.2]_D
U 1158, 0000,00FC,0380,0800,0000,1010  ;2002  TRAP.FPA   ; TRAP FPA TO 0
U 1010, 0818,0039,2D80,0800,0000,115A  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 1011, 0000,003C,5980,3C00,0000,1159  ;2004  ID[ACC.2]_D
U 1159, 0000,00FE,0380,0800,0000,0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
            ;2006 ;
            ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2008 ;%0
            ;2009 ;$ 0000.0000. 0000.0000
            ;2010 ;
            ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2014 ;
U 115A. 0000.003C.65F8.0800.0084.715B  ;2015 GENTRA: SC_K[.10],Q_0
U 115B. 0D00.003C.8D80.0800.0084.715C  ;2016  D_DAL.SC,SC_K[.1F]
U 115C. 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2018 ;
            ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2021 ;
            ;2022 =0
U 1012. 0F00.003D.0180.0800.0000.115A  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 1013. 0000.003C.5980.3C00.0000.115D  ;2024  ID[ACC.2]_D
U 115D. 0000.00FC.0380.0800.0000.115E  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 115E. 0000.003C.0180.0800.0000.115F  ;2026  NOP
U 115F. 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
            ;2028 ;
            ;2029 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2030 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2031 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2032 ;
U 1160. 2000.003C.0180.0800.0000.1161  ;2033 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 1161. 3000.003C.0180.6000.0000.1162  ;2034  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 1162. 0000.003C.0180.F800.0000.1163  ;2035 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1163. 0000.003C.0180.F800.0000.1164  ;2036  MCT/ALLOW.IB.READ
U 1164. 0000.003C.0180.F800.0000.1165  ;2037  MCT/ALLOW.IB.READ
U 1165. 0000.003C.0180.F800.0000.1166  ;2038  MCT/ALLOW.IB.READ
U 1166. 0000.003C.0180.F800.0000.1167  ;2039  MCT/ALLOW.IB.READ
U 1167. 1000.003C.0180.F800.0000.1168  ;2040  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1168. 0000.003E.0180.0800.0000.0001  ;2041  RETURN1
U 1169. 3000.003C.0180.0800.0000.1162  ;2042 WAITIB: IBC/START,J/IBW
            ;2043 ;
            ;2044 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
            ;2045 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
            ;2046 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
            ;2047 ; RC[8]  A OPERAND BITS <63:32>
            ;2048 ; RC[7]  A OPERAND BITS <31:00>
            ;2049 ; RC[6]  B OPERAND BITS <63:32>
            ;2050 ; RC[5]  B OPERAND BITS <31:00>
            ;2051 ; RC[0A]  RESULT BITS <63:32>
            ;2052 ; RC[9]  RESULT BITS <31:00>
            ;2053 ;
U 116A. 0810.0038.0180.0938.0000.116B  ;2054 ADDSIM: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 116B. 0010.0038.01C0.0928.0000.116C  ;2055  Q_RC[5]
U 116C. 001D.0014.0180.09C8.0010.116D  ;2056  ALU_D+Q,RC[9]_ALU,CLK,UBCC ; AND SAVE THE CARRY
U 116D. 0810.1B38.0180.0940.0000.100E  ;2057  D_RC[8],ALU.C?
U 100E. 0000.003C.0180.0800.0000.116E  ;2058 =1110 J/ADSM1   ; NO CARRY
U 100F. 0819.0014.0580.0800.0000.116E  ;2059  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
            ;2060     ; RESULT
```

```
U 116E, 0010.0038.01C0.0930.0000.116F  ;2061 ADSM1: Q_RC[6]
U 116F, 001D.0016.0180.09D0.0000.0001  ;2062  ALU_D+Q,RC[0A]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
          ;2063      ; AND RETURN.
          ;2064 ;
          ;2065 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU SUBTRACT. IT PERFORMS A
          ;2066 ; DOUBLE PRECISION SUB ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
          ;2067 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
          ;2068 ; RC[8]  A OPERAND BITS <63:32>
          ;2069 ; RC[7]  A OPERAND BITS <31:00>
          ;2070 ; RC[6]  B OPERAND BITS <63:32>
          ;2071 ; RC[5]  B OPERAND BITS <31:00>
          ;2072 ;
U 1170, 0010.0038.01C0.0938.0000.1171  ;2073 SUBSIM: Q_RC[7]   ; DO LO SUB
U 1171, 0810.0038.0180.0928.0000.1172  ;2074  D_RC[5]
U 1172, 001D.0000.0180.09C8.0010.1173  ;2075  ALU_D-Q,RC[9]_ALU,CLK.UBCC ; SAVE BORROW
U 1173, 0010.1B38.01C0.0940.0000.101E  ;2076  Q_RC[8],ALU.C?
U 101E, 0019.2014.05C0.0800.0000.101F  ;2077 =1110 ALU_Q+K[.1],Q_ALU ; SUB BORROW
U 101F, 0810.0038.0180.0930.0000.1174  ;2078 SBSM1: D_RC[6]
U 1174, 001D.0002.0180.09D0.0000.0001  ;2079  ALU_D-Q,RC[0A]_ALU,RETURN1 ; DO UPPER HALF OF SUB.
          ;2080 ;
          ;2081 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
          ;2082 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], R[0D]
          ;2083 ; AND R[0C] AND LEAVES THE RESULT IN RC[8] AND RC[7].
          ;2084 ; RC[8]  A OPERAND BITS <63:32>
          ;2085 ; RC[7]  A OPERAND BITS <31:00>
          ;2086 ; R[0C]  B OPERAND BITS <63:32>
          ;2087 ; R[0D]  B OPERAND BITS <31:00>
          ;2088 ; RC[8]  RESULT BITS <63:32>
          ;2089 ; RC[7]  RESULT BITS <31:00>
          ;2090 ;
U 1175, 0810.0038.0180.0938.0000.1176  ;2091 INCA: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 1176, 0000.003C.01C0.0A68.0000.1177  ;2092  Q_R[0D]
U 1177, 001D.0014.0180.09B8.0010.1178  ;2093  ALU_D+Q,RC[7]_ALU,CLK.UBCC ; AND SAVE THE CARRY
U 1178, 0810.1B38.0180.0940.0000.102E  ;2094  D_RC[8],ALU.C?
U 102E, 0000.003C.0180.0800.0000.1179  ;2095 =1110 J/INCA1   ; NO CARRY
U 102F, 0819.0014.0580.0800.0000.1179  ;2096  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
          ;2097      ; RESULT
U 1179, 0000.003C.01C0.0A60.0000.117A  ;2098 INCA1: Q_R[0C]
U 117A, 001D.0016.0180.09C0.0000.0001  ;2099  ALU_D+Q,RC[8]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
          ;2100      ; AND RETURN.
          ;2101 ;
          ;2102 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
          ;2103 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[5], RC[6], R[0D]
          ;2104 ; AND R[0C] AND LEAVES THE RESULT IN RC[6] AND RC[5].
          ;2105 ; RC[6]  A OPERAND BITS <63:32>
          ;2106 ; RC[5]  A OPERAND BITS <31:00>
          ;2107 ; R[0C]  B OPERAND BITS <63:32>
          ;2108 ; R[0D]  B OPERAND BITS <31:00>
          ;2109 ; RC[6]  RESULT BITS <63:32>
          ;2110 ; RC[5]  RESULT BITS <31:00>
          ;2111 ;
U 117B, 0810.0038.0180.0928.0000.117C  ;2112 INCB: D_RC[5]   ; FIRST ADD LO FRACTIONS
U 117C, 0000.003C.01C0.0A68.0000.117D  ;2113  Q_R[0D]
U 117D, 0019.2024.59C0.0800.0000.117E  ;2114  ALU_Q[ANDNOT]K[.7F],Q_ALU
U 117E, 001D.0014.0180.09A8.0010.117F  ;2115  ALU_D+Q,RC[5]_ALU,CLK.UBCC ; AND SAVE THE CARRY
```

```
U 117F. 0810.1B38.0180.0930.0000.103E  ;2116  D_RC[6],ALU.C?
U 103E. 0000.003C.0180.0800.0000.1180  ;2117 =1110 J/INCB1    ; NO CARRY
U 103F. 0819.0014.0580.0800.0000.1180  ;2118  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
          ;2119      ; RESULT
U 1180. 0000.003C.01C0.0A60.0000.1181  ;2120 INCB1: Q_R[0C]
U 1181. 001D.0016.0180.09B0.0000.0001  ;2121  ALU_D+Q,RC[6]_ALU,RETURN1   ; DO HIGH HALF OF THE ADD
          ;2122      ; AND RETURN.
          ;2123 ;
          ;2124 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2125 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2126 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2127 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2128 ;
U 1182. 0000.003C.01C0.0A28.0000.1183  ;2129 UNSEQA: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1183. 0800.003C.B980.0A20.0084.7184  ;2130  D_R[4],SC_K[.19]
U 1184. 0D00.003C.0180.0800.0000.1185  ;2131  D_DAL.SC
U 1185. 0000.003C.65E0.0800.0084.7186  ;2132  Q_D,SC_K[.10]
U 1186. 0D00.003C.0180.0800.0000.1187  ;2133  D_DAL.SC
U 1187. 0001.003C.0180.0AB8.0000.1188  ;2134  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 1188. 0800.003C.01C0.0A20.0000.1189  ;2135  D_R[4],Q_R[4]
U 1189. 0819.0024.5980.0800.0000.118A  ;2136  ALU_D[ANDNOT]K[.7F],D_ALU
U 118A. 0100.003C.5D88.0800.0084.718B  ;2137  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 118B. 0D00.003C.0180.0800.0000.118C  ;2138  D_DAL.SC
U 118C. 0001.003C.0180.0AB0.0084.718D  ;2139  R[6]_D,SC_K[.8]
U 118D. 0818.0034.59F8.0A28.0000.118E  ;2140  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 118E. 0D00.003C.01C0.0A30.0000.118F  ;2141  D_DAL.SC,Q_R[6]
U 118F. 001D.0032.0180.0AB0.0000.0001  ;2142  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2143 ;
          ;2144 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2145 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2146 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2147 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2148 ;
U 1190. 0000.003C.01C0.0A28.0000.1191  ;2149 UNSEQB: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1191. 0800.003C.B980.0A20.0084.7192  ;2150  D_R[4],SC_K[.19]
U 1192. 0D00.003C.0180.0800.0000.1193  ;2151  D_DAL.SC
U 1193. 0000.003C.65E0.0800.0084.7194  ;2152  Q_D,SC_K[.10]
U 1194. 0D00.003C.0180.0800.0000.1195  ;2153  D_DAL.SC
U 1195. 0001.003C.0180.0AB8.0000.1196  ;2154  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 1196. 0800.003C.01C0.0A20.0000.1197  ;2155  D_R[4],Q_R[4]
U 1197. 0819.0024.5980.0800.0000.1198  ;2156  ALU_D[ANDNOT]K[.7F],D_ALU
U 1198. 0100.003C.5D88.0800.0084.7199  ;2157  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 1199. 0D00.003C.0180.0800.0000.119A  ;2158  D_DAL.SC
U 119A. 0001.003C.0180.0AB0.0084.719B  ;2159  R[6]_D,SC_K[.8]
U 119B. 0818.0034.59F8.0A28.0000.119C  ;2160  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 119C. 0D00.003C.01C0.0A30.0000.119D  ;2161  D_DAL.SC,Q_R[6]
U 119D. 001D.0032.0180.0AB0.0000.0001  ;2162  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2163 ;
          ;2164 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          ;2165 ; IN RC[0A] AND BITS <31:00> IN RC[9]) INTO THE FLOATING POINT FORMAT
          ;2166 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN RC[0E]
          ;2167 ; AND BITS <31:00> IN RC[0D].
          ;2168 ;
U 119E. 0810.0038.0180.0950.0000.119F  ;2169 UNSRES: D_RC[0A]
U 119F. 0010.0038.01C0.0948.0000.11A0  ;2170  Q_RC[9]
```

```
U 11A0. 0001.003C.0180.0AA0.0000.1014  ;2171  R[4]_D
U 1014. 0001.203D.0180.0AA8.0000.1182  ;2172 =0 R[5]_Q,CALL,J/UNSEQA
U 1015. 0800.003C.0180.0A30.0000.11A1  ;2173  D_R[6]
U 11A1. 0000.003C.01C0.0A38.0000.11A2  ;2174  Q_R[7]
U 11A2. 0001.003C.0180.09F0.0000.11A3  ;2175  RC[0E]_D
U 11A3. 0001.203E.0180.09E8.0000.0001  ;2176  RC[0D]_Q,RETURN1
          :2177 ;
          :2178 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          :2179 ; IN RC[8] AND BITS <31:00> IN RC[7]) INTO THE FLOATING POINT FORMAT
          :2180 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[8]
          :2181 ; AND BITS <31:00> IN R[9].
          :2182 ;
U 11A4. 0810.0038.0180.0940.0000.11A5  ;2183 UNSAOP: D_RC[8]
U 11A5. 0010.0038.01C0.0938.0000.11A6  ;2184  Q_RC[7]
U 11A6. 0001.003C.0180.0AA0.0000.1016  ;2185  R[4]_D
U 1016. 0001.203D.0180.0AA8.0000.1182  ;2186 =0 R[5]_Q,CALL,J/UNSEQA
U 1017. 0800.003C.0180.0A30.0000.11A7  ;2187  D_R[6]
U 11A7. 0000.003C.01C0.0A38.0000.11A8  ;2188  Q_R[7]
U 11A8. 0001.003C.0180.0AC0.0000.11A9  ;2189  R[8]_D
U 11A9. 0001.203E.0180.0AC8.0000.0001  ;2190  R[9]_Q,RETURN1
          :2191 ;
          :2192 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          :2193 ; IN RC[6] AND BITS <31:00> IN RC[5]) INTO THE FLOATING POINT FORMAT
          :2194 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[0A]
          :2195 ; AND BITS <31:00> IN R[0B].
          :2196 ;
U 11AA. 0810.0038.0180.0930.0000.11AB  ;2197 UNSBOP: D_RC[6]
U 11AB. 0010.0038.01C0.0928.0000.11AC  ;2198  Q_RC[5]
U 11AC. 0001.003C.0180.0AA0.0000.1018  ;2199  R[4]_D
U 1018. 0001.203D.0180.0AA8.0000.1182  ;2200 =0 R[5]_Q,CALL,J/UNSEQA
U 1019. 0800.003C.0180.0A30.0000.11AD  ;2201  D_R[6]
U 11AD. 0000.003C.01C0.0A38.0000.11AE  ;2202  Q_R[7]
U 11AE. 0001.003C.0180.0AD0.0000.11AF  ;2203  R[0A]_D
U 11AF. 0001.203E.0180.0AD8.0000.0001  ;2204  R[0B]_Q,RETURN1
          :2205 ;
          :2206 ; THIS ROUTINE IS CALLED BY THE FALU TESTS TO LOAD LA, SA,
          :2207 ; LB, SB, AR AND BR, FALU/A.B FUNCTION IS EXECUTED
          :2208 ; AND THE RESULTS RETRIEVED.
          :2209 ;
U 11B0. 0818.0038.3180.0800.0000.101A  ;2210 EXEC: D_K[.40]
U 101A. 0819.0031.6180.0800.0000.115A  ;2211 =0 ALU_D[OR]K[.F],D_ALU,CALL,J/GENTRA
U 101B. 0000.003C.5980.3C00.0000.11B1  ;2212  ID[ACC.2]_D
U 11B1. 0000.00FC.0380.0800.0000.11B2  ;2213  TRAP,FPA  ; TRAP TO FPA 4F
U 11B2. 0000.003C.0180.0A10.0000.11B3  ;2214  LAB_R[2]  ; AT FPA 4F
U 11B3. 0000.007C.0180.0A10.0000.11B4  ;2215  LAB_R[2],CPSYNC  ; AT FPA 4F
U 11B4. 0000.003C.0180.0A18.0000.11B5  ;2216  LAB_R[3]  ; AT FPA 39
U 11B5. 0000.007C.0180.0A18.0000.11B6  ;2217  LAB_R[3],CPSYNC  ; AT FPA 39
U 11B6. 0000.003C.0180.0A40.0000.11B7  ;2218  LAB_R[8]  ; AT FPA 3A
U 11B7. 0000.007C.0180.0A40.0000.11B8  ;2219  LAB_R[8],CPSYNC  ; AT FPA 3A
U 11B8. 0000.003C.0180.0A48.0000.11B9  ;2220  LAB_R[9]  ; AT FPA 3B
U 11B9. 0000.007C.0180.0A48.0000.11BA  ;2221  LAB_R[9],CPSYNC  ; AT FPA 3B
U 11BA. 0000.003C.0180.0A50.0000.11BB  ;2222  LAB_R[0A]  ; AT FPA 3C
U 11BB. 0000.007C.0180.0A50.0000.11BC  ;2223  LAB_R[0A],CPSYNC ; AT FPA 3C
U 11BC. 0000.003C.0180.0A58.0000.11BD  ;2224  LAB_R[0B]  ; AT FPA 3D
U 11BD. 0000.007C.0180.0A58.0000.11BE  ;2225  LAB_R[0B],CPSYNC ; AT FPA 3D
```

```
U 11BE, 0A00.003C.0180.0800.0000.11BF  ;2226  D_ACC    ; AT FPA 3E, GET RESULT HI
U 11BF, 0000.003C.01D8.0800.0000.11C0  ;2227  Q_ACC    ; AT FPA 3F, GET RESULT LO
U 11C0, 0001.003C.0180.09E0.0000.11C1  ;2228  RC[0C]_D
U 11C1, 0001.203E.0180.09D8.0000.0001  ;2229  RC[0B]_Q,RETURN1
```

```
                ;2230  .PAGE "TEST 263 EALU/A+B TEST"
                ;2231  ;*****************************************************************
                ;2232  ;+
                ;2233  ; EALU/A+B TEST
                ;2234  ;
                ;2235  ; TEST DESCRIPTION
                ;2236  ; THIS IS A TEST OF THE EALU FUNCTION A+B. EVERY POSSIBLE
                ;2237  ; COMBINATION OF A OPERANDS WITH B OPERANDS IS TRIED. THE
                ;2238  ; RESULT IS CHECKED IN TWO WAYS:
                ;2239  ;  1 IF THE RESULT WAS AN OVERFLOW (BIT 8 OF THE
                ;2240  ;    EALU IS SET) THE RESULT IS STORED IN THE XR.
                ;2241  ;    THE BUS SHOULD BE FORCED TO 0'S BECAUSE OF
                ;2242  ;    EALU<8>=1. THEN THE ACTUAL RESULT
                ;2243  ;    OF THE ADDITION IS TESTED BY READING THE XR
                ;2244  ;    BACK THROUGH THE EALU.
                ;2245  ;  2 IF THE RESULT WAS NOT OVERFLOW THEN IT IS
                ;2246  ;    READ BACK DIRECTLY ONTO THE BUS AND CHECKED.
                ;2247  ;
                ;2248  ; FPA UCODE USED
                ;2249  ; LOCATION CONTENTS
                ;2250  ; 90:  BSC/R,SCR/CPU,EAC/LDA,NAD/91,WAIT
                ;2251  ; 91:  BSC/R,SCR/CPU,EAC/LDB,NAD/92,WAIT
                ;2252  ; 92:  AMXC/LA,BMXC/LB,EALUC/A+B,BSC/NH,EAC/LDXR,NAD/93
                ;2253  ; 93:  BSC/R,SCR/CPU,EAC/LDAB,NAD/94,WAIT
                ;2254  ; 94:  AMXC/LA,BMXC/XR,EALUC/A+B,BSC/NH,NAD/94
                ;2255  ;
                ;2256  ; LOGIC DESCRIPTION
                ;2257  ;
                ;2258  ; ERROR DESCRIPTION
                ;2259  ; EXPECTED RESULT, IN BITS <14:07>
                ;2260  ; GOT RESULT, IN BITS <14:07>
                ;2261  ; CONTENTS OF LA, IN BITS <14:07>>
                ;2262  ; CONTENTS OF LB, IN BITS <14:07>>
                ;2263  ;
                ;2264  ; SYNC POINT DESCRIPTION
                ;2265  ;
                ;2266  ;--
                ;2267  ;*****************************************************************
                ;2268  =0
U 101C. 0000,003D,0180,0800,0000,1138   ;2269 EAPLB: NEWTST
U 101D. 0018,0038,1180,09F8,0000,11C2   ;2270  RC[0F]_K[.4]
U 11C2. 0018,0038,4980,0A80,0000,11C3   ;2271  R[0]_K[.FF]  ; SET UP THE COUNTERS WHICH ARE
U 11C3. 0818,0038,4980,0A88,0000,11C4   ;2272  R[1]_K[.FF],D_K[.FF] ; ALSO THE OPERANDS.
U 11C4. 0019,0014,0580,0A90,0000,11C5   ;2273  ALU_D+K[.1],R[2]_ALU
U 11C5. 0000,003C,5D80,0800,0084,71C6   ;2274  SC_K[.7]
U 11C6. 0803,0010,6180,0800,0084,71C7   ;2275  ALU_MASK+1,D_ALU,SC_K[.F]
U 11C7. 0003,0010,01C0,0800,0000,11C8   ;2276  ALU_MASK+1,Q_ALU
U 11C8. 001D,0024,01C0,0A98,0000,11C9   ;2277  ALU_D[ANDNOT]Q,Q_ALU,R[3]_ALU
U 11C9. 0003,003C,0180,0AA0,0000,1020   ;2278  R[4]_0
U 1020. 0000,003D,0180,0800,0000,114A   ;2279 =0 ERLOOP
U 1021. 0000,003C,0180,0800,0000,1022   ;2280  NOP
                ;2281  =0
U 1022. 0000,003D,0180,0800,0000,1154   ;2282 EAPLB1: CALL,J/FPINIT
U 1023. 0800,003C,5DF8,0A00,0084,71CA   ;2283  SC_K[.7],D_R[0],Q_0 ; GENERATE THE LA OPERAND.
U 11CA. 0D00,003C,0180,0800,0000,11CB   ;2284  D_DAL.SC
```

```
U 11CB, 0001,003C,0180,0AA8,0000,11CC  ;2285  R[5]_D
U 11CC, 0001,003C,0180,09E0,0000,11CD  ;2286  RC[0C]_D
U 11CD, 0800,003C,0180,0A08,0000,11CE  ;2287  D_R[1]   ; GENERATE THE LB OPERAND
U 11CE, 0D00,003C,0180,0800,0000,11CF  ;2288  D_DAL.SC
U 11CF, 0001,003C,0180,0AB0,0000,11D0  ;2289  R[6]_D
U 11D0, 0001,003C,0180,09D8,0000,11D1  ;2290  RC[0B]_D
U 11D1, 0818,0038,C580,0800,0000,1024  ;2291  D_K[.88]
U 1024, 0819,0015,0180,0800,0000,115A  ;2292 =0 ALU_D+K[.8],D_ALU,CALL,J/GENTRA ; GENERATE THE ADDRESS OF
      ;2293     ; THE FPA UCODE USED.
U 1025, 0000,003C,5980,3C00,0000,11D2  ;2294  ID[ACC.2]_D
U 11D2, 0000,00FC,0380,0800,0000,11D3  ;2295  TRAP.FPA
U 11D3, 0000,003C,0180,0A28,0000,11D4  ;2296  LAB_R[5]  ; AT FPA 90
U 11D4, 0000,007C,0180,0A28,0000,11D5  ;2297  LAB_R[5],CPSYNC  ; AT FPA 90
U 11D5, 0000,003C,0180,0A30,0000,11D6  ;2298  LAB_R[6]  ; AT FPA 91
U 11D6, 0000,007C,0180,0A30,0000,11D7  ;2299  LAB_R[6],CPSYNC  ; AT FPA 91
U 11D7, 0000,003C,01D8,0800,0000,11D8  ;2300  Q_ACC   ; AT FPA 92, GET RESULT
U 11D8, 0000,003C,0180,0A20,0000,11D9  ;2301  LAB_R[4]  ; AT FPA 93
U 11D9, 0000,007C,0180,0A20,0000,11DA  ;2302  LAB_R[4],CPSYNC  ; AT FPA 93
U 11DA, 0A00,003C,0180,0800,0000,11DB  ;2303  D_ACC
U 11DB, 080D,0034,0180,0A18,0000,11DC  ;2304  ALU_D[AND]R[3],D_ALU ; GET JUST EXPONENT
U 11DC, 0001,003C,0180,0AC0,0000,11DD  ;2305  R[8]_D
U 11DD, 000D,2034,01C0,0A18,0000,11DE  ;2306  ALU_Q[AND]R[3],Q_ALU
U 11DE, 0001,203C,0180,0AC8,0000,11DF  ;2307  R[9]_Q
U 11DF, 0800,003C,0180,0A00,0000,11E0  ;2308  D_R[0]   ; CALCULATE THE EXPECTED RESULT
U 11E0, 0000,003C,01C0,0A08,0000,11E1  ;2309  Q_R[1]
U 11E1, 081D,0014,0180,0AD0,0000,11E2  ;2310  ALU_D+Q,D_ALU,R[0A]_ALU
U 11E2, 000D,0034,0180,0A10,0010,11E3  ;2311  ALU_D[AND]R[2],CLK.UBCC ; EALU<8> SET?
U 11E3, 0000,013C,0180,0800,0000,1026  ;2312  Z?
U 1026, 0000,003C,0180,0800,0000,11EF  ;2313 =0 J/EAPLB3
U 1027, 0800,003C,0180,0A50,0000,11E4  ;2314 EAPLB2: D_R[0A]  ; CHECK THE RESULT
U 11E4, 0819,0034,4980,0800,0000,11E5  ;2315  ALU_D[AND]K[.FF],D_ALU
U 11E5, 0000,003C,5D80,0800,0084,71E6  ;2316  SC_K[.7]
U 11E6, 0D00,003C,0180,0800,0000,11E7  ;2317  D_DAL.SC
U 11E7, 0001,003C,0180,09F0,0000,11E8  ;2318  RC[0E]_D
U 11E8, 0000,003C,01C0,0A40,0000,11E9  ;2319  Q_R[8]
U 11E9, 0001,203C,0180,09E8,0000,11EA  ;2320  RC[0D]_Q
U 11EA, 001D,0000,0180,0800,0010,11EB  ;2321  ALU_D-Q,CLK.UBCC
U 11EB, 0000,013C,0180,0800,0000,1028  ;2322  Z?
U 1028, 0000,003D,0180,0800,0000,114B  ;2323 =0 ERROR1   ; EALU OUTPUT INCORRECT.
U 1029, 0000,003C,0180,0A08,0010,11EC  ;2324  ALU_R[1],CLK.UBCC ; TRIED ALL LA OPERNDS?
U 11EC, 0000,013C,0180,0800,0000,102A  ;2325  Z?
U 102A, 0018,0000,0580,0A88,0000,1022  ;2326 =0 R[1]_LA-K[.1],J/EAPLB1 ; IF NOT CONTINUE.
U 102B, 0018,0038,4980,0A88,0000,11ED  ;2327  R[1]_K[.FF]  ; ELSE RESET A OPERAND AND
U 11ED, 0000,003C,0180,0A00,0010,11EE  ;2328  ALU_R[0],CLK.UBCC ; SEE IF ALL LB OPERANDS HAVE BEEN
U 11EE, 0000,013C,0180,0800,0000,102C  ;2329  Z?   ; TRIED.
U 102C, 0018,0000,0580,0A80,0000,1022  ;2330 =0 R[0]_LA-K[.1],J/EAPLB1
U 102D, 0000,003C,0180,0800,0000,11F3  ;2331  J/EAPLBD  ; FINISHED TEST.
U 11EF, 0003,003C,0180,09F0,0000,11F0  ;2332 EAPLB3: RC[0E]_0
U 11F0, 0800,003C,0180,0A48,0000,11F1  ;2333  D_R[9]   ; EALU<8>=1 SO RESULT SHOULD BE 0.
U 11F1, 0001,003C,0180,09E8,0010,11F2  ;2334  RC[0D]_D,CLK.UBCC
U 11F2, 0000,013C,0180,0800,0000,1030  ;2335  Z?
U 1030, 0000,003D,0180,0800,0000,114B  ;2336 =0 ERROR1   ; EXPONENT HAS NOT 0 WITH
      ;2337     ; EALU<8>=1
U 1031, 0000,003C,0180,0800,0000,1027  ;2338  J/EAPLB2
U 11F3, 0000,003C,0180,0800,0000,1032  ;2339 EAPLBD: NOP
```

```
                ;2340  .PAGE "TEST 264 EALU/A-B TEST"
                ;2341 ;*****************************************************************
                ;2342 ;++
                ;2343 ; EALU/A-B TEST
                ;2344 ;
                ;2345 ; TEST DESCRIPTION
                ;2346 ; THIS IS A TEST OF THE EALU FUNCTION A-B. EVERY POSSIBLE
                ;2347 ; COMBINATION OF A OPERANDS WITH B OPERANDS IS TRIED. THE
                ;2348 ; RESULT IS CHECKED IN TWO WAYS:
                ;2349 ;  1 IF THE RESULT WAS AN UNDERFLOW (BIT 8 OF THE
                ;2350 ;    EALU IS SET) THE RESULT IS STORED IN THE XR.
                ;2351 ;    THE BUS SHOULD BE FORCED TO 0'S BECAUSE OF
                ;2352 ;    EALU<8>=1. THEN THE ACTUAL RESULT
                ;2353 ;    OF THE ADDITION IS TESTED BY READING THE XR
                ;2354 ;    BACK THROUGH THE EALU.
                ;2355 ;  2 IF THE RESULT WAS NOT UNDERFLOW THEN IT IS
                ;2356 ;    READ BACK DIRECTLY ONTO THE BUS AND CHECKED.
                ;2357 ;
                ;2358 ; FPA UCODE USED
                ;2359 ; LOCATION CONTENTS
                ;2360 ; 95:  BSC/R,SCR/CPU,EAC/LDA,NAD/96,WAIT
                ;2361 ; 96:  BSC/R,SCR/CPU,EAC/LDB,NAD/97,WAIT
                ;2362 ; 97:  AMXC/LA,BMXC/LB,EALUC/A+B,BSC/NH,EAC/LDXR,NAD/98
                ;2363 ; 98:  BSC/R,SCR/CPU,EAC/LDAB,NAD/99,WAIT
                ;2364 ; 99:  AMXC/LA,BMXC/XR,EALUC/A+B,BSC/NH,NAD/99
                ;2365 ;
                ;2366 ; LOGIC DESCRIPTION
                ;2367 ;
                ;2368 ; ERROR DESCRIPTION
                ;2369 ; EXPECTED RESULT, IN BITS <14:07>
                ;2370 ; GOT RESULT, IN BITS <14:07>
                ;2371 ; CONTENTS OF LA, IN BITS <14:07>>
                ;2372 ; CONTENTS OF LB, IN BITS <14:07>>
                ;2373 ;
                ;2374 ; SYNC POINT DESCRIPTION
                ;2375 ;
                ;2376 ;--
                ;2377 ;*****************************************************************
                ;2378 =0
U 1032,  0000.003D.0180.0800.0000.1138  ;2379 EASUB: NEWTST
U 1033,  0018.0038.1180.09F8.0000.11F4  ;2380  RC[0F]_K[.4]
U 11F4,  0018.0038.4980.0A80.0000.11F5  ;2381  R[0]_K[.FF]  ; SET UP THE COUNTERS WHICH ARE
U 11F5,  0818.0038.4980.0A88.0000.11F6  ;2382  R[1]_K[.FF],D_K[.FF] ; ALSO THE OPERANDS.
U 11F6,  0019.0014.0580.0A90.0000.11F7  ;2383  ALU_D+K[.1],R[2]_ALU
U 11F7,  0000.003C.5D80.0800.0084.71F8  ;2384  SC_K[.7]
U 11F8,  0803.0010.6180.0800.0084.71F9  ;2385  ALU_MASK+1,D_ALU,SC_K[.F]
U 11F9,  0003.0010.01C0.0800.0000.11FA  ;2386  ALU_MASK+1,Q_ALU
U 11FA,  001D.0024.01C0.0A98.0000.11FB  ;2387  ALU_D[ANDNOT]Q,Q_ALU,R[3]_ALU
U 11FB,  0003.003C.0180.0AA0.0000.1034  ;2388  R[4]_0
U 1034,  0000.003D.0180.0800.0000.114A  ;2389 =0 ERLOOP
U 1035,  0000.003C.0180.0800.0000.1036  ;2390  NOP
                ;2391 =0
U 1036,  0000.003D.0180.0800.0000.1154  ;2392 EASUB1: CALL,J/FPINIT
U 1037,  0800.003C.5DF8.0A00.0084.71FC  ;2393  SC_K[.7],D_R[0],Q_0 ; GENERATE THE LA OPERAND.
U 11FC,  0D00.003C.0180.0800.0000.11FD  ;2394  D_DAL.SC
```

```
U 11FD. 0001.003C.0180.0AA8.0000.11FE  ;2395  R[5]_D
U 11FE. 0001.003C.0180.09E0.0000.11FF  ;2396  RC[0C]_D
U 11FF. 0800.003C.0180.0A08.0000.1200  ;2397  D_R[1]   ; GENERATE THE LB OPERAND
U 1200. 0D00.003C.0180.0800.0000.1201  ;2398  D_DAL.SC
U 1201. 0001.003C.0180.0AB0.0000.1202  ;2399  R[6]_D
U 1202. 0001.003C.0180.09D8.0000.1203  ;2400  RC[0B]_D
U 1203. 0818.0038.C580.0800.0000.1204  ;2401  D_K[.88]
U 1204. 0819.0014.0580.0800.0000.1205  ;2402  ALU_D+K[.1],D_ALU
U 1205. 0819.0014.1180.0800.0000.1038  ;2403  ALU_D+K[.4],D_ALU
U 1038. 0819.0015.0180.0800.0000.115A  ;2404 =0 ALU_D+K[.8],D_ALU,CALL,J/GENTRA ; GENERATE THE ADDRESS OF
        ;2405     ; THE FPA UCODE USED.
U 1039. 0000.003C.5980.3C00.0000.1206  ;2406  ID[ACC.2]_D
U 1206. 0000.00FC.0380.0800.0000.1207  ;2407  TRAP.FPA
U 1207. 0000.003C.0180.0A28.0000.1208  ;2408  LAB_R[5]  ; AT FPA 95
U 1208. 0000.007C.0180.0A28.0000.1209  ;2409  LAB_R[5],CPSYNC  ; AT FPA 95
U 1209. 0000.003C.0180.0A30.0000.120A  ;2410  LAB_R[6]  ; AT FPA 96
U 120A. 0000.007C.0180.0A30.0000.120B  ;2411  LAB_R[6],CPSYNC  ; AT FPA 96
U 120B. 0000.003C.01D8.0800.0000.120C  ;2412  Q_ACC   ; AT FPA 97, GET RESULT
U 120C. 0000.003C.0180.0A20.0000.120D  ;2413  LAB_R[4]  ; AT FPA 98
U 120D. 0000.007C.0180.0A20.0000.120E  ;2414  LAB_R[4],CPSYNC  ; AT FPA 98
U 120E. 0A00.003C.0180.0800.0000.120F  ;2415  D_ACC
U 120F. 080D.0034.0180.0A18.0000.1210  ;2416  ALU_D[AND]R[3],D_ALU ; GET JUST EXPONENT
U 1210. 0001.003C.0180.0AC0.0000.1211  ;2417  R[8]_D
U 1211. 000D.2034.01C0.0A18.0000.1212  ;2418  ALU_Q[AND]R[3],Q_ALU
U 1212. 0001.203C.0180.0AC8.0000.1213  ;2419  R[9]_Q
U 1213. 0800.003C.0180.0A00.0000.1214  ;2420  D_R[0]   ; CALCULATE THE EXPECTED RESULT
U 1214. 0000.003C.01C0.0A08.0000.1215  ;2421  Q_R[1]
U 1215. 081D.0000.0180.0AD0.0000.1216  ;2422  ALU_D-Q,D_ALU,R[0A]_ALU
U 1216. 000D.0034.0180.0A10.0010.1217  ;2423  ALU_D[AND]R[2],CLK.UBCC ; EALU<8> SET?
U 1217. 0000.013C.0180.0800.0000.103A  ;2424  Z?
U 103A. 0000.003C.0180.0800.0000.1223  ;2425 =0 J/EASUB3
U 103B. 0800.003C.0180.0A50.0000.1218  ;2426 EASUB2: D_R[0A]  ; CHECK THE RESULT
U 1218. 0819.0034.4980.0800.0000.1219  ;2427  ALU_D[AND]K[.FF],D_ALU
U 1219. 0000.003C.5D80.0800.0084.721A  ;2428  SC_K[.7]
U 121A. 0D00.003C.0180.0800.0000.121B  ;2429  D_DAL.SC
U 121B. 0001.003C.0180.09F0.0000.121C  ;2430  RC[0E]_D
U 121C. 0000.003C.01C0.0A40.0000.121D  ;2431  Q_R[8]
U 121D. 0001.203C.0180.09E8.0000.121E  ;2432  RC[0D]_Q
U 121E. 001D.0000.0180.0800.0010.121F  ;2433  ALU_D-Q,CLK.UBCC
U 121F. 0000.013C.0180.0800.0000.103C  ;2434  Z?
U 103C. 0000.003D.0180.0800.0000.114B  ;2435 =0 ERROR1   ; EALU OUTPUT INCORRECT.
U 103D. 0000.003C.0180.0A08.0010.1220  ;2436  ALU_R[1],CLK.UBCC ; TRIED ALL LA OPERNDS?
U 1220. 0000.013C.0180.0800.0000.1040  ;2437  Z?
U 1040. 0018.0000.0580.0A88.0000.1036  ;2438 =0 R[1]_LA-K[.1],J/EASUB1 ; IF NOT CONTINUE.
U 1041. 0018.0038.4980.0A88.0000.1221  ;2439  R[1]_K[.FF]  ; ELSE RESET A OPERAND AND
U 1221. 0000.003C.0180.0A00.0010.1222  ;2440  ALU_R[0],CLK.UBCC ; SEE IF ALL LB OPERANDS HAVE BEEN
U 1222. 0000.013C.0180.0800.0000.1042  ;2441  Z?   ; TRIED.
U 1042. 0018.0000.0580.0A80.0000.1036  ;2442 =0 R[0]_LA-K[.1],J/EASUB1
U 1043. 0000.003C.0180.0800.0000.1227  ;2443  J/EASUBD  ; FINISHED TEST.
U 1223. 0003.003C.0180.09F0.0000.1224  ;2444 EASUB3: RC[0E]_0
U 1224. 0800.003C.0180.0A48.0000.1225  ;2445  D_R[9]   ; EALU<8>=1 SO RESULT SHOULD BE 0.
U 1225. 0001.003C.0180.09E8.0010.1226  ;2446  RC[0D]_D,CLK.UBCC
U 1226. 0000.013C.0180.0800.0000.1044  ;2447  Z?
U 1044. 0000.003D.0180.0800.0000.114B  ;2448 =0 ERROR1   ; EXPONENT WAS NOT 0 WITH
        ;2449     ; EALU<8>=1
```

```
U 1045. 0000.003C.0180.0800.0000.103B  ;2450  J/EASUB2
U 1227. 0000.003C.0180.0800.0000.1046  ;2451 EASUBD: NOP
```

```
;2452  .PAGE "TEST 265 BEN/3 AND BEN/5 BIT 2 TEST"
;2453 ;*****************************************************************
;2454 ;*
;2455 ; BEN/3 AND BEN/5 BIT 2 TEST
;2456 ;
;2457 ; TEST DESCRIPTION
;2458 ; THIS TEST LOADS LA, LB, SA AND SB WITH VARIOUS PATTERNS
;2459 ; AND THEN EXECUTES BEN/3 OR BEN/5 IN THE FPA.
;2460 ; BEN/3 BIT 2 RSV OPR H
;2461 ; BEN/3 BIT 1 EXPB=0 AND SB=0
;2462 ; BEN/3 BIT 0 EXPA=0 AND SA=0
;2463 ; BEN/5 BIT 2 LA OR LB IS 0
;2464 ;
;2465 ;    SUBTEST LA LB SA SB BEN/3 BEN/5 (WITH NAD/3)
;2466 ; 1 XX 00 0 0 2 7
;2467 ; 2 00 XX 0 0 1 7
;2468 ; 3 XX XX 1 1 0 3
;2469 ; 4 00 00 0 0 3 NT
;2470 ; 5 00 00 1 0 4 NT
;2471 ; 6 00 00 0 1 4 NT
;2472 ; 7 XX 00 1 0 2 NT
;2473 ; 8 00 XX 0 1 1 NT
;2474 ;
;2475 ; THE ABOVE TABLE IDENTIFIES THE CONTENTS OF LA, LB, SA AND
;2476 ; SB USED IN EACH SUBTEST. NOTE THAT XX IS A PATTERN GENERATED
;2477 ; BY FLOATING A 1 THRU A FIELD OF 0'S. THE TABLE ALSO
;2478 ; IDENTIFIES THE EXPECTED NAD PRODUCED WHEN THE BEN IS EXECUTED.
;2479 ; HERE NT MEANS THAT BEN/5 IS NOT TESTED WITH THIS COMBINATION
;2480 ; OF OPERANDS.
;2481 ;
;2482 ; FPA UCODE USED
;2483 ; LOCATION CONTENTS
;2484 ; 30:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,NAD/31,WAIT
;2485 ; 31:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,NAD/32,WAIT
;2486 ; 32:  BEN/3,NAD/0
;2487 ; 33:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,NAD/34,WAIT
;2488 ; 34:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,NAD/35,WAIT
;2489 ; 35:  BEN/5,NAD/3
;2490 ;
;2491 ; LOGIC DESCRIPTION
;2492 ;
;2493 ; ERROR DESCRIPTION
;2494 ; EXPECTED NEXT ADDRESS, GENERATED BY BEN
;2495 ; GOT NEXT ADDRESS
;2496 ; CONTENTS OF LA, BITS <07:00>
;2497 ; CONTENTS OF SA, 0 OR 1
;2498 ; CONTENTS OF LB, BITS <07:00>
;2499 ; CONTENTS OF SB, 0 OR 1
;2500 ;
;2501 ; SYNC POINT DESCRIPTION
;2502 ;
;2503 ;--
;2504 ;*****************************************************************
;2505 =0
U 1046, 0000,003D,0180,0800,0000,1138  ;2506 B35T: NEWTST
```

```
U 1047, 0003.003C.0180.0A80.0000.1228  ;2507  R[0]_0
U 1228, 0018.0038.4580.0A88.0000.1048  ;2508  R[1]_K[.8000]
U 1048, 0818.0039.7980.0800.0000.115A  ;2509 =0 D_K[.30],CALL,J/GENTRA ; GENERATE THE ADDRESSES OF
U 1049, 0001.003C.0180.0AB0.0000.1229  ;2510  R[6]_D    ; FPA ROUTINES USED.
U 1229, 0818.0038.7980.0800.0000.104C  ;2511  D_K[.30]
U 104C, 0819.0015.0D80.0800.0000.115A  ;2512 =0 ALU_D+K[.3],D_ALU,CALL,J/GENTRA
U 104D, 0001.003C.0180.0AB8.0000.122A  ;2513  R[7]_D
U 122A, 0018.0038.5D80.0AC0.0000.1050  ;2514  R[8]_K[.7]
U 1050, 0000.003D.0180.0800.0000.114A  ;2515 =0 ERLOOP
U 1051, 0018.0038.D580.09F8.0000.122B  ;2516 B35TA: RC[0F]_K[.6]
U 122B, 0000.003C.0180.0A40.0082.122C  ;2517  SC_R[8]    ; GENERATE THE FLOATING 1
U 122C, 0003.001C.5D80.0AA8.0084.922D  ;2518  ALU_NOT.MASK,R[5]_ALU,SC_SC+K[.7] ; TEST PATTERN.
U 122D, 0803.001C.0180.0A90.0000.122E  ;2519  ALU_NOT.MASK,R[2]_ALU,D_ALU
U 122E, 0019.0030.4580.0A98.0000.1052  ;2520  ALU_D[OR]K[.8000],R[3]_ALU
          :2521 ;////////////////////////////////////////////////////////////////
          :2522 ;+
          :2523 ; SUBTEST 1
          :2524 ; SA=0, LA=FLOATING PATTERN, SB=0, LB=0
          :2525 ;-
U 1052, 0000.003D.0180.0800.0000.114F  ;2526 =0 SUBTEST
U 1053, 0003.003C.0180.09D8.0000.122F  ;2527  RC[0B]_0   ; SET UP ERROR TYPE OUT
U 122F, 0003.003C.0180.09C8.0000.1230  ;2528  RC[9]_0    ; IN CASE OF FAILURE.
U 1230, 0003.003C.0180.09D0.0000.1231  ;2529  RC[0A]_0
U 1231, 0000.003C.0180.0A28.0000.1054  ;2530  LAB_R[5]
U 1054, 0000.003D.0180.09E0.0000.1154  ;2531 =0 RC[0C]_LA,CALL,J/FPINIT ; INITIALIZE THE FPA.
U 1055, 0800.003C.0180.0A30.0000.1232  ;2532  D_R[6]
U 1232, 0000.003C.5980.3C00.0000.1233  ;2533  ID[ACC.2]_D
U 1233, 0018.00F8.0B80.09F0.0000.1234  ;2534  TRAP.FPA,RC[0E]_K[.2] ; TRAP TO FPA 30.
U 1234, 0000.003C.0180.0A00.0000.1235  ;2535  LAB_R[0]   ; AT FPA 30
U 1235, 0000.007C.0180.0A00.0000.1236  ;2536  LAB_R[0],CPSYNC
U 1236, 0000.003C.0180.0A10.0000.1237  ;2537  LAB_R[2]   ; AT FPA 31
U 1237, 0000.007C.0180.0A10.0000.1238  ;2538  LAB_R[2],CPSYNC
U 1238, 0000.003C.59F0.2C00.0000.1239  ;2539  Q_ID[ACC.2]   ; AT FPA 32, GET NAD.
U 1239, 0019.2034.81C0.09E8.0000.123A  ;2540  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 123A, 0019.2000.0980.0800.0010.123B  ;2541  ALU_Q-K[.2],CLK.UBCC ; CHECK THE RESULT.
U 123B, 0000.013C.0180.0800.0000.1056  ;2542  Z?
U 1056, 0000.003D.0180.0800.0000.114B  ;2543 =0 ERROR1    ; BEN/3 FAILED.
U 1057, 0000.003C.0180.0800.0000.1058  ;2544  NOP
U 1058, 0000.003D.0180.0800.0000.1154  ;2545 =0 CALL,J/FPINIT   ; INITIALIZE THE FPA.
U 1059, 0800.003C.0180.0A38.0000.123C  ;2546  D_R[7]
U 123C, 0000.003C.5980.3C00.0000.123D  ;2547  ID[ACC.2]_D
U 123D, 0818.00F8.5F80.09F0.0000.123E  ;2548  TRAP.FPA,RC[0E]_K[.7],D_K[.7] ; TRAP TO FPA 33.
U 123E, 0000.003C.0180.0A00.0000.123F  ;2549  LAB_R[0]   ; AT FPA 33
U 123F, 0000.007C.0180.0A00.0000.1240  ;2550  LAB_R[0],CPSYNC
U 1240, 0000.003C.0180.0A10.0000.1241  ;2551  LAB_R[2]   ; AT FPA 34
U 1241, 0000.007C.0180.0A10.0000.1242  ;2552  LAB_R[2],CPSYNC
U 1242, 0000.003C.59F0.2C00.0000.1243  ;2553  Q_ID[ACC.2]   ; AT FPA 34, GET NAD.
U 1243, 0019.2034.81C0.09E8.0000.1244  ;2554  ALU_Q[AND]K[.3FF],RC[0D]_ALU,Q_ALU
U 1244, 001D.0000.0180.0800.0010.1245  ;2555  ALU_D-Q,CLK.UBCC ; CHECK RESULT
U 1245, 0000.013C.0180.0800.0000.105C  ;2556  Z?
U 105C, 0000.003D.0180.0800.0000.114B  ;2557 =0 ERROR1    ; BEN/7 FAILED.
U 105D, 0000.003C.0180.0800.0000.1060  ;2558  NOP
          :2559 ;////////////////////////////////////////////////////////////////
          :2560 ;+
          :2561 ; SUBTEST 2
```

```
            ;2562 ; SA=0, LA=0, SB=0, LB=FLOATING PATTERN
            ;2563 ;-
U 1060, 0000.003D.0180.0800.0000.114F  ;2564 =0 SUBTEST
U 1061, 0003.003C.0180.09E0.0000.1246  ;2565  RC[0C]_0  ; SET UP DATA IN CASE OF ERROR.
U 1246, 0000.003C.0180.0A28.0000.1062  ;2566  LAB_R[5]
U 1062, 0000.003D.0180.09D0.0000.1154  ;2567 =0 RC[0A]_LA,CALL,J/FPINIT ; INITIALIZE THE FPA
U 1063, 0800.003C.0180.0A30.0000.1247  ;2568  D_R[6]
U 1247, 0000.003C.5980.3C00.0000.1248  ;2569  ID[ACC.2]_D
U 1248, 0018.00F8.0780.09F0.0000.1249  ;2570  TRAP.FPA,RC[0E]_K[.1] ; TRAP TO FPA 30
U 1249, 0000.003C.0180.0A10.0000.124A  ;2571  LAB_R[2]  ; AT FPA 30
U 124A, 0000.007C.0180.0A10.0000.124B  ;2572  LAB_R[2],CPSYNC
U 124B, 0000.003C.0180.0A00.0000.124C  ;2573  LAB_R[0]  ; AT FPA 31
U 124C, 0000.007C.0180.0A00.0000.124D  ;2574  LAB_R[0],CPSYNC
U 124D, 0000.003C.59F0.2C00.0000.124E  ;2575  Q_ID[ACC.2]  ; AT FPA 32, GET NAD.
U 124E, 0019.2034.81C0.09E8.0000.124F  ;2576  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 124F, 0019.2000.0580.0800.0010.1250  ;2577  ALU_Q-K[.1],CLK.UBCC ; CHECK THE RESULT
U 1250, 0000.013C.0180.0800.0000.1064  ;2578  Z?
U 1064, 0000.003D.0180.0800.0000.114B  ;2579 =0 ERROR1   ; BEN/3 FAILED
U 1065, 0000.003C.0180.0800.0000.1066  ;2580  NOP
U 1066, 0000.003D.0180.0800.0000.1154  ;2581 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1067, 0800.003C.0180.0A38.0000.1251  ;2582  D_R[7]
U 1251, 0000.003C.5980.3C00.0000.1252  ;2583  ID[ACC.2]_D
U 1252, 0818.00F8.5F80.09F0.0000.1253  ;2584  TRAP.FPA,RC[0E]_K[.7],D_K[.7] ; TRAP TO FPA 33.
U 1253, 0000.003C.0180.0A10.0000.1254  ;2585  LAB_R[2]  ; AT FPA 33
U 1254, 0000.007C.0180.0A10.0000.1255  ;2586  LAB_R[2],CPSYNC
U 1255, 0000.003C.0180.0A00.0000.1256  ;2587  LAB_R[0]  ; AT FPA 34
U 1256, 0000.007C.0180.0A00.0000.1257  ;2588  LAB_R[0],CPSYNC
U 1257, 0000.003C.59F0.2C00.0000.1258  ;2589  Q_ID[ACC.2]  ; AT FPA 35, GET NAD.
U 1258, 0019.2034.81C0.09E8.0000.1259  ;2590  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1259, 001D.0000.0180.0800.0010.125A  ;2591  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
U 125A, 0000.013C.0180.0800.0000.1068  ;2592  Z?
U 1068, 0000.003D.0180.0800.0000.114B  ;2593 =0 ERROR1   ; BEN/5 FAILED.
U 1069, 0000.003C.0180.0800.0000.106C  ;2594  NOP
            ;2595 ;////////////////////////////////////////////////////////////////
            ;2596 ;*
            ;2597 ; SUBTEST 3
            ;2598 ; SA=1, LA=FLOATING PATTERN, SB=1, LB=FLOATING PATTERN
            ;2599 ;-
U 106C, 0000.003D.0180.0800.0000.114F  ;2600 =0 SUBTEST
U 106D, 0018.0038.0580.09D8.0000.125B  ;2601  RC[0B]_K[.1]  ; SET UP DATA IN CASE OF ERROR.
U 125B, 0018.0038.0580.09C8.0000.125C  ;2602  RC[9]_K[.1]
U 125C, 0000.003C.0180.0A28.0000.125D  ;2603  LAB_R[5]
U 125D, 0000.003C.0180.09E0.0000.1070  ;2604  RC[0C]_LA
U 1070, 0000.003D.0180.09D0.0000.1154  ;2605 =0 RC[0A]_LA,CALL,J/FPINIT ; INITIALIZE THE FPA
U 1071, 0800.003C.0180.0A30.0000.125E  ;2606  D_R[6]
U 125E, 0000.003C.5980.3C00.0000.125F  ;2607  ID[ACC.2]_D
U 125F, 0003.00FC.0380.09F0.0000.1260  ;2608  TRAP.FPA,RC[0E]_0 ; TRAP TO FPA 30
U 1260, 0000.003C.0180.0A18.0000.1261  ;2609  LAB_R[3]  ; AT FPA 30
U 1261, 0000.007C.0180.0A18.0000.1262  ;2610  LAB_R[3],CPSYNC
U 1262, 0000.003C.0180.0A18.0000.1263  ;2611  LAB_R[3]  ; AT FPA 31
U 1263, 0000.007C.0180.0A18.0000.1264  ;2612  LAB_R[3],CPSYNC
U 1264, 0000.003C.59F0.2C00.0000.1265  ;2613  Q_ID[ACC.2]  ; AT FPA 32, GET NAD.
U 1265, 0019.2034.8180.09E8.0010.1266  ;2614  ALU_Q[AND]K[.3FF],RC[0D]_ALU,CLK.UBCC ; CHECK THE RESULT.
U 1266, 0000.013C.0180.0800.0000.1072  ;2615  Z?
U 1072, 0000.003D.0180.0800.0000.114B  ;2616 =0 ERROR1   ; BEN/3 FAILED.
```

```
U 1073. 0000.003C.0180.0800.0000.1074  ;2617  NOP
U 1074. 0000.003D.0180.0800.0000.1154  ;2618 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA.
U 1075. 0800.003C.0180.0A38.0000.1267  ;2619  D_R[7]
U 1267. 0000.003C.5980.3C00.0000.1268  ;2620  ID[ACC.2]_D
U 1268. 0018.00F8.0F80.09F0.0000.1269  ;2621  TRAP.FPA,RC[0E]_K[.3] ; TRAP TO FPA 33
U 1269. 0000.003C.0180.0A18.0000.126A  ;2622  LAB_R[3]  ; AT FPA 33
U 126A. 0000.007C.0180.0A18.0000.126B  ;2623  LAB_R[3],CPSYNC
U 126B. 0000.003C.0180.0A18.0000.126C  ;2624  LAB_R[3]  ; AT FPA 34
U 126C. 0000.007C.0180.0A18.0000.126D  ;2625  LAB_R[3],CPSYNC
U 126D. 0000.003C.59F0.2C00.0000.126E  ;2626  Q_ID[ACC.2]  ; AT FPA 35, GET NAD.
U 126E. 0019.2034.81C0.09E8.0000.126F  ;2627  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 126F. 0019.2000.0D80.0800.0010.1270  ;2628  ALU_Q-K[.3],CLK.UBCC ; CHECK THE RESULT
U 1270. 0000.013C.0180.0800.0000.1076  ;2629  Z?
U 1076. 0000.003D.0180.0800.0000.114B  ;2630 =0 ERROR1   ; BEN/5 FAILED.
U 1077. 0000.003C.0180.0800.0000.1078  ;2631  NOP
        ;2632 ;////////////////////////////////////////////////////////////
        ;2633 ;*
        ;2634 ; SUBTEST 4
        ;2635 ; SA=0, LA=0, SB=0, LB=0
        ;2636 ;-
U 1078. 0000.003D.0180.0800.0000.114F  ;2637 =0 SUBTEST
U 1079. 0003.003C.0180.09E0.0000.1271  ;2638  RC[0C]_0
U 1271. 0003.003C.0180.09D8.0000.1272  ;2639  RC[0B]_0
U 1272. 0003.003C.0180.09D0.0000.107C  ;2640  RC[0A]_0
U 107C. 0003.003D.0180.09C8.0000.1154  ;2641 =0 RC[9]_0,CALL,J/FPINIT
U 107D. 0800.003C.0180.0A30.0000.1273  ;2642  D_R[6]
U 1273. 0000.003C.5980.3C00.0000.1274  ;2643  ID[ACC.2]_D
U 1274. 0018.00F8.0F80.09F0.0000.1275  ;2644  TRAP.FPA,RC[0E]_K[.3] ; TRAP TO FPA 30.
U 1275. 0000.003C.0180.0A00.0000.1276  ;2645  LAB_R[0]  ; AT FPA 30
U 1276. 0000.007C.0180.0A00.0000.1277  ;2646  LAB_R[0],CPSYNC
U 1277. 0000.003C.0180.0A00.0000.1278  ;2647  LAB_R[0]  ; AT FPA 31
U 1278. 0000.007C.0180.0A00.0000.1279  ;2648  LAB_R[0],CPSYNC
U 1279. 0000.003C.59F0.2C00.0000.127A  ;2649  Q_ID[ACC.2]  ; AT FPA 32, GET NAD.
U 127A. 0019.2034.81C0.09E8.0000.127B  ;2650  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 127B. 0019.2000.0D80.0800.0010.127C  ;2651  ALU_Q-K[.3],CLK.UBCC ; CHECK THE RESULT.
U 127C. 0000.013C.0180.0800.0000.1080  ;2652  Z?
U 1080. 0000.003D.0180.0800.0000.114B  ;2653 =0 ERROR1   ; BEN/3 FAILED
U 1081. 0000.003C.0180.0800.0000.1082  ;2654  NOP
        ;2655 ;////////////////////////////////////////////////////////////
        ;2656 ;*
        ;2657 ; SUBTEST 5
        ;2658 ; SA=1, LA=0, SB=0, LB=0
        ;2659 ;-
U 1082. 0000.003D.0180.0800.0000.114F  ;2660 =0 SUBTEST
U 1083. 0018.0038.0580.09D8.0000.1084  ;2661  RC[0B]_K[.1]
U 1084. 0000.003D.0180.0800.0000.1154  ;2662 =0 CALL,J/FPINIT  ; INITIALIZE THE FPA
U 1085. 0800.003C.0180.0A30.0000.127D  ;2663  D_R[6]
U 127D. 0000.003C.5980.3C00.0000.127E  ;2664  ID[ACC.2]_D
U 127E. 0018.00F8.1380.09F0.0000.127F  ;2665  TRAP.FPA,RC[0E]_K[.4] ; TRAP TO FPA 30.
U 127F. 0000.003C.0180.0A00.0000.1280  ;2666  LAB_R[0]  ; AT FPA 30
U 1280. 0000.007C.0180.0A00.0000.1281  ;2667  LAB_R[0],CPSYNC
U 1281. 0000.003C.0180.0A08.0000.1282  ;2668  LAB_R[1]  ; AT FPA 31
U 1282. 0000.007C.0180.0A08.0000.1283  ;2669  LAB_R[1],CPSYNC
U 1283. 0000.003C.59F0.2C00.0000.1284  ;2670  Q_ID[ACC.2]  ; AT FPA 32, GET NAD.
U 1284. 0019.2034.81C0.09E8.0000.1285  ;2671  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
```

```
U 1285. 0019.2000.1180.0800.0010.1286  ;2672  ALU_Q-K[.4],CLK.UBCC ; CHECK THE RESULT.
U 1286. 0000.013C.0180.0800.0000.1086  ;2673  Z?
U 1086. 0000.003D.0180.0800.0000.114B  ;2674 =0 ERROR1    ; BEN/3 FAILED.
U 1087. 0000.003C.0180.0800.0000.1088  ;2675  NOP
          ;2676 ;////////////////////////////////////////////////////////////
          ;2677 ;*
          ;2678 ; SUBTEST 6
          ;2679 ; SA=0, LA=0, SB=1, LB=0
          ;2680 ;-
U 1088. 0000.003D.0180.0800.0000.114F  ;2681 =0 SUBTEST
U 1089. 0003.003C.0180.09D8.0000.108A  ;2682  RC[0B]_0
U 108A. 0018.0039.0580.09C8.0000.1154  ;2683 =0 RC[9]_K[.1],CALL,J/FPINIT ; INITIALIZE THE FPA.
U 108B. 0800.003C.0180.0A30.0000.1287  ;2684  D_R[6]
U 1287. 0000.003C.5980.3C00.0000.1288  ;2685  ID[ACC.2]_D
U 1288. 0018.00F8.1380.09F0.0000.1289  ;2686  TRAP.FPA,RC[0E]_K[.4] ; TRAP TO FPA 30.
U 1289. 0000.003C.0180.0A08.0000.128A  ;2687  LAB_R[1]   ; AT FPA 30
U 128A. 0000.007C.0180.0A08.0000.128B  ;2688  LAB_R[1],CPSYNC
U 128B. 0000.003C.0180.0A00.0000.128C  ;2689  LAB_R[0]   ; AT FPA 31
U 128C. 0000.007C.0180.0A00.0000.128D  ;2690  LAB_R[0],CPSYNC
U 128D. 0000.003C.59F0.2C00.0000.128E  ;2691  Q_ID[ACC.2]   ; AT FPA 32, GET NAD.
U 128E. 0019.2034.81C0.09E8.0000.128F  ;2692  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 128F. 0019.2000.1180.0800.0010.1290  ;2693  ALU_Q-K[.4],CLK.UBCC ; CHECK THE RESULT.
U 1290. 0000.013C.0180.0800.0000.108C  ;2694  Z?
U 108C. 0000.003D.0180.0800.0000.114B  ;2695 =0 ERROR1    ; BEN/3 FAILED.
U 108D. 0000.003C.0180.0800.0000.108E  ;2696  NOP
          ;2697 ;////////////////////////////////////////////////////////////
          ;2698 ;*
          ;2699 ; SUBTEST 7
          ;2700 ; SA=1, LA=FLOATING PATTERN, SB=0, LB=0
          ;2701 ;-
U 108E. 0000.003D.0180.0800.0000.114F  ;2702 =0 SUBTEST
U 108F. 0018.0038.0580.09D8.0000.1291  ;2703  RC[0B]_K[.1]
U 1291. 0003.003C.0180.09C8.0000.1292  ;2704  RC[9]_0
U 1292. 0000.003C.0180.0A28.0000.1090  ;2705  LAB_R[5]
U 1090. 0000.003D.0180.09E0.0000.1154  ;2706 =0 RC[0C]_LA,CALL,J/FPINIT ; INITIALIZE THE FPA.
U 1091. 0800.003C.0180.0A30.0000.1293  ;2707  D_R[6]
U 1293. 0000.003C.5980.3C00.0000.1294  ;2708  ID[ACC.2]_D
U 1294. 0018.00F8.0B80.09F0.0000.1295  ;2709  TRAP.FPA,RC[0E]_K[.2] ; TRAP TO FPA 30.
U 1295. 0000.003C.0180.0A00.0000.1296  ;2710  LAB_R[0]   ; AT FPA 30
U 1296. 0000.007C.0180.0A00.0000.1297  ;2711  LAB_R[0],CPSYNC
U 1297. 0000.003C.0180.0A18.0000.1298  ;2712  LAB_R[3]   ; AT FPA 31
U 1298. 0000.007C.0180.0A18.0000.1299  ;2713  LAB_R[3],CPSYNC
U 1299. 0000.003C.59F0.2C00.0000.129A  ;2714  Q_ID[ACC.2]   ; AT FPA 32, GET NAD.
U 129A. 0019.2034.81C0.09E8.0000.129B  ;2715  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 129B. 0019.2000.0980.0800.0010.129C  ;2716  ALU_Q-K[.2],CLK.UBCC ; CHECK THE RESULT.
U 129C. 0000.013C.0180.0800.0000.1092  ;2717  Z?
U 1092. 0000.003D.0180.0800.0000.114B  ;2718 =0 ERROR1    ; BEN/3 FAILED.
U 1093. 0000.003C.0180.0800.0000.1094  ;2719  NOP
          ;2720 ;////////////////////////////////////////////////////////////
          ;2721 ;*
          ;2722 ; SUBTEST 8
          ;2723 ; SA=0, LA=0, SB=1, LB=FLOATING PATTERN
          ;2724 ;-
U 1094. 0000.003D.0180.0800.0000.114F  ;2725 =0 SUBTEST
U 1095. 0003.003C.0180.09E0.0000.129D  ;2726  RC[0C]_0
```

```
U 129D. 0003.003C.0180.09D8.0000.129E  ;2727  RC[0B]_0
U 129E. 0018.0038.0580.09C8.0000.129F  ;2728  RC[9]_K[.1]
U 129F. 0000.003C.0180.0A28.0000.1096  ;2729  LAB_R[5]
U 1096. 0000.003D.0180.09D0.0000.1154  ;2730 =0 RC[0A]_LA,CALL,J/FPINIT ; INITIALIZE THE FPA.
U 1097. 0800.003C.0180.0A30.0000.12A0  ;2731  D_R[6]
U 12A0. 0000.003C.5980.3C00.0000.12A1  ;2732  ID[ACC.2]_D
U 12A1. 0018.00F8.0780.09F0.0000.12A2  ;2733  TRAP.FPA,RC[0E]_K[.1] ; TRAP TO FPA 30.
U 12A2. 0000.003C.0180.0A18.0000.12A3  ;2734  LAB_R[3]  ; AT FPA 30
U 12A3. 0000.007C.0180.0A18.0000.12A4  ;2735  LAB_R[3].CPSYNC
U 12A4. 0000.003C.0180.0A00.0000.12A5  ;2736  LAB_R[0]  ; AT FPA 31
U 12A5. 0000.007C.0180.0A00.0000.12A6  ;2737  LAB_R[0].CPSYNC
U 12A6. 0000.003C.59F0.2C00.0000.12A7  ;2738  Q_ID[ACC.2]  ; AT FPA 32, GET NAD.
U 12A7. 0019.2034.81C0.09E8.0000.12A8  ;2739  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12A8. 0019.2000.0580.0800.0010.12A9  ;2740  ALU_Q-K[.1],CLK.UBCC ; CHECK THE RESULT.
U 12A9. 0000.013C.0180.0800.0000.1098  ;2741  Z?
U 1098. 0000.003D.0180.0800.0000.114B  ;2742 =0 ERROR1
          ;2743 ; SEE IF ALL PATTERNS HAVE BEEN TRIED:
U 1099. 0000.003C.0180.0A40.0010.12AB  ;2744  ALU_R[8],CLK.UBCC
U 12AB. 0000.013C.0180.0800.0000.109A  ;2745  Z?
U 109A. 0018.0000.0580.0AC0.0000.1051  ;2746 =0 R[8]_LA-K[.1],J/B35TA ; NO SO CONTINUE
U 109B. 0000.003C.0180.0800.0000.109C  ;2747  NOP   ; ELSE DONE.
```

```
                ;2748  .PAGE "TEST 266 ASHF AND EXPONENT DIFFERENCE TEST 1"
                ;2749 ;*****************************************************************
                ;2750 ;++
                ;2751 ; ASHF AND EXPONENT DIFFERENCE TEST 1
                ;2752 ;
                ;2753 ; TEST DESCRIPTION
                ;2754 ; THIS TEST SETS BIT 63 IN AR AND CLEARS BR AND THEN DOES A FALU
                ;2755 ; A+B WITH EXPONENT DIFFENCES OF 63 THRU 0.
                ;2756 ;
                ;2757 ; LOGIC DESCRIPTION
                ;2758 ;
                ;2759 ; FPA UCODE USED
                ;2760 ; LOCATION CONTENTS
                ;2761 ; 40:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/LD,WAIT,NAD/41
                ;2762 ; 41:  FADC/R1,BSC/FAL.HL,NAD/PSTALL
                ;2763 ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
                ;2764 ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
                ;2765 ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
                ;2766 ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
                ;2767 ;
                ;2768 ; ERROR DESCRIPTION
                ;2769 ; EXPECTED RESULT, FALU HI ON BUS A
                ;2770 ; EXPECTED RESULT, FALU LO ON BUS A
                ;2771 ; GOT RESULT HI
                ;2772 ; GOT RESULT LO
                ;2773 ; CONTENTS OF AR DURING SHIFT HI
                ;2774 ; CONTENTS OF AR DURING SHIFT LO
                ;2775 ; EXPECTED OUTPUT OF ASHF AND FALU HI
                ;2776 ; EXPECTED OUTPUT OF ASHF AND FALU LO
                ;2777 ; CONTENTS OF LA DURING SHIFT <07:00>
                ;2778 ; CONTENTS OF LB DURING SHIFT <07:00>
                ;2779 ; EXPONENT DIFFERENCE (LB-LA) IN BITS <07:00>
                ;2780 ;
                ;2781 ; SYNC POINT DESCRIPTION
                ;2782 ;
                ;2783 ;--
                ;2784 ;*****************************************************************
                ;2785 =0
U 109C, 0000.003D.0180.0800.0000.1138  ;2786 ASHF1: NEWTST
U 109D, 0818.0038.8580.0800.0000.12AC  ;2787  D_K[.C]
U 12AC, 0019.0000.0580.09F8.0000.12AD  ;2788  ALU_D-K[.1],RC[0F]_ALU
U 12AD, 0018.0038.41F8.0AC0.0000.12AE  ;2789  R[8]_K[.80],Q_0  ; INITIALIZE SOME CONSTANTS
U 12AE, 0000.003C.03B0.0800.0000.12AF  ;2790  Q_Q.RIGHT,SI/7
U 12AF, 0001.203C.0180.09D0.0000.12B0  ;2791  RC[0A]_Q
U 12B0, 0003.003C.0180.09C8.0000.12B1  ;2792  RC[9]_0
U 12B1, 0018.0038.0580.09B0.0000.12B2  ;2793  RC[6]_K[.1]
U 12B2, 0018.0038.5580.0A90.0000.109E  ;2794  R[2]_K[.3F]  ; INITIALIZE A COUNTER
U 109E, 0000.003D.0180.0800.0000.114A  ;2795 =0 ERLOOP
U 109F, 0000.003C.0180.0800.0000.10A0  ;2796  NOP
                ;2797 =0
U 10A0, 0000.003D.0180.0800.0000.1154  ;2798 ASHF1A: CALL,J/FPINIT
U 10A1, 0018.0038.2180.0800.0200.10A2  ;2799  VA_K[.14]  ; LOAD ADDD INSTRUCTIO
U 10A2, 0000.003D.0180.0800.0000.1160  ;2800 =0 CALL,J/LOADIB  ; INTO THE IB.
U 10A3, 0818.0038.3180.0800.0000.10A4  ;2801  D_K[.40]  ; GENERATE ADDRESS OF THE FPA
U 10A4, 0000.003D.0180.0800.0000.115A  ;2802 =0 CALL,J/GENTRA  ; UCODE USED.
```

```
U 10A5. 0000.003C.5980.3C00.0000.12B3  ;2803  ID[ACC.2]_D
U 12B3. 0000.00FC.0380.0800.0000.12B4  ;2804  TRAP.FPA  ; TRAP TO FPA 40
U 12B4. 0000.003C.0180.0A40.0000.12B5  ;2805  LAB_R[8]  ; AT FPA 40
U 12B5. 0000.007C.0180.0A40.0000.12B6  ;2806  LAB_R[8],CPSYNC  ; AT FPA 40
U 12B6. 0000.003C.0180.0800.0000.12B7  ;2807  NOP  ; AT FPA 41, AR BIT 63 SET AND
      ;2808     ; BR CLEARED HERE.
U 12B7. 0000.003C.5DC0.0A10.0084.72B8  ;2809  Q_R[2],SC_K[.7]  ; COMPUTE THE LA AND LB VALUES
U 12B8. 0818.0038.5580.0800.0000.12B9  ;2810  D_K[.3F]  ; APPROPRIATE FOR THIS SHIFT COUNT.
U 12B9. 081D.0000.0180.09A0.0000.12BA  ;2811  ALU_D-Q,D_ALU,RC[04]_ALU
U 12BA. 0819.0014.05F8.09A8.0000.12BB  ;2812  ALU_D+K[.1],D_ALU,RC[05]_ALU,Q_0
U 12BB. 0D00.003C.0180.0800.0000.12BC  ;2813  D_DAL.SC
U 12BC. 0001.003C.0180.0AD8.0000.12BD  ;2814  R[0B]_D
U 12BD. 0818.0038.3180.0800.0000.10A6  ;2815  D_K[.40]
U 10A6. 0819.0015.0980.0800.0000.115A  ;2816 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; GENERATE ADDRESS OF FPA UCODE.
U 10A7. 0000.003C.5980.3C00.0000.12BE  ;2817  ID[ACC.2]_D
U 12BE. 0000.00FC.0380.0800.0000.12BF  ;2818  TRAP.FPA  ; TRAP TO FPA 42
U 12BF. 0000.003C.0180.0A40.0000.12C0  ;2819  LAB_R[8]  ; AT FPA 42
U 12C0. 0000.007C.0180.0A40.0000.12C1  ;2820  LAB_R[8],CPSYNC  ; AT FPA 42
U 12C1. 0000.003C.0180.0A58.0000.12C2  ;2821  LAB_R[0B]  ; AT FPA 43
U 12C2. 0000.007C.0180.0A58.0000.12C3  ;2822  LAB_R[0B],CPSYNC  ; AT FPA 43
U 12C3. 0000.003C.01D8.0800.0000.12C4  ;2823  Q_ACC  ; AT FPA 44, GET RESULT HI
U 12C4. 0A00.003C.0180.0800.0000.12C5  ;2824  D_ACC  ; AT FPA 45, GET RESULT LO
U 12C5. 0001.203C.0180.09E0.0000.12C6  ;2825  RC[0C]_Q
U 12C6. 0001.003C.0180.09D8.0000.12C7  ;2826  RC[0B]_D
U 12C7. 0000.003C.0180.0A10.0082.12C8  ;2827  SC_R[2]  ; COMPUTE EXPECTED RESULT.
U 12C8. 0810.0038.0180.0920.0000.12C9  ;2828  D_RC[04]
U 12C9. 0019.0034.7580.0800.0010.12CA  ;2829  ALU_D[AND]K[.20],CLK.UBCC
U 12CA. 0000.013C.0180.0800.0000.10A8  ;2830  Z?
U 10A8. 0000.003C.0180.0800.0000.12CC  ;2831 =0 J/ASHF1B
U 10A9. 0803.0010.0180.0800.0000.12CB  ;2832  ALU_MASK+1,D_ALU
U 12CB. 0000.003C.01F8.0800.0000.12CE  ;2833  Q_0,J/ASHF1C
U 12CC. 081B.0000.0580.0800.0000.12CD  ;2834 ASHF1B: ALU_0-K[.1],D_ALU
U 12CD. 0003.0010.01C0.0800.0000.12CE  ;2835  ALU_MASK+1,Q_ALU
U 12CE. 0001.003C.0180.09C0.0000.12CF  ;2836 ASHF1C: RC[8]_D
U 12CF. 0001.203C.0180.09B8.0000.12D0  ;2837  RC[7]_Q
U 12D0. 0001.003C.0180.0AA0.0000.10AA  ;2838  R[4]_D
U 10AA. 0001.203D.0180.0AA8.0000.1182  ;2839 =0 R[5]_Q,CALL,J/UNSEQA
U 10AB. 0800.003C.0180.0A30.0000.12D1  ;2840  D_R[6]
U 12D1. 0000.003C.01C0.0A38.0000.12D2  ;2841  Q_R[7]
U 12D2. 0001.003C.0180.09F0.0000.12D3  ;2842  RC[0E]_D
U 12D3. 0001.203C.0180.09E8.0000.12D4  ;2843  RC[0D]_Q
U 12D4. 0010.0038.01C0.0960.0000.12D5  ;2844  Q_RC[0C]
U 12D5. 001D.0000.0180.0800.0010.12D6  ;2845  ALU_D-Q,CLK.UBCC ; CHECK RESULT
U 12D6. 0000.013C.0180.0800.0000.10AC  ;2846  Z?
U 10AC. 0000.003D.0180.0800.0000.114B  ;2847 =0 ERROR1  ; RESULT HI INCORRECT.
U 10AD. 0810.0038.0180.0968.0000.12D7  ;2848  D_RC[0D]
U 12D7. 0010.0038.01C0.0958.0000.12D8  ;2849  Q_RC[0B]
U 12D8. 001D.0000.0180.0800.0010.12D9  ;2850  ALU_D-Q,CLK.UBCC
U 12D9. 0000.013C.0180.0800.0000.10AE  ;2851  Z?
U 10AE. 0000.003D.0180.0800.0000.114B  ;2852 =0 ERROR1  ; RESULT LO INCORRECT.
U 10AF. 0000.003C.0180.0A10.0010.12DA  ;2853  ALU_R[2],CLK.UBCC ; TRIED ALL SHIFT COUNTS?
U 12DA. 0000.013C.0180.0800.0000.10B0  ;2854  Z?
U 10B0. 0018.0000.0580.0A90.0000.10A0  ;2855 =0 R[2]_LA-K[.1],J/ASHF1A ; NO SO CONTINUE TEST.
U 10B1. 0000.003C.0180.0800.0000.10B2  ;2856  NOP
      ;2857 ;
```

```
;2858 ;%14
;2859 ;$ 5060,0050
;2860 ;
```

```
            ;2861  .PAGE "TEST 267 ASHF AND EXPONENT DIFFERENCE TEST 2"
            ;2862 ;*****************************************************************
            ;2863 ;**
            ;2864 ; ASHF AND EXPONENT DIFFERENCE TEST 2
            ;2865 ;
            ;2866 ; TEST DESCRIPTION
            ;2867 ; THIS TEST SETS JUST BIT 62 IN THE AR AND BR AND THEN
            ;2868 ; DOES A FALU A+B FUNCTION WITH EXPONENTS SUCH THAT
            ;2869 ; AR IS SHIFTED RIGHT WITH A SHIFT COUNT OF 63 THRU 0.
            ;2870 ;
            ;2871 ; LOGIC DESCRIPTION
            ;2872 ;
            ;2873 ; FPA UCODE USED
            ;2874 ; LOCATION CONTENTS
            ;2875 ; 46:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/LD,WAIT,NAD/PSTALL
            ;2876 ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
            ;2877 ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
            ;2878 ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
            ;2879 ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
            ;2880 ;
            ;2881 ; ERROR DESCRIPTION
            ;2882 ; EXPECTED RESULT, FALU HI ON BUS A
            ;2883 ; EXPECTED RESULT, FALU LO ON BUS A
            ;2884 ; GOT RESULT HI
            ;2885 ; GOT RESULT LO
            ;2886 ; CONTENTS OF AR DURING SHIFT HI
            ;2887 ; CONTENTS OF AR DURING SHIFT LO
            ;2888 ; EXPECTED OUTPUT OF ASHF AND FALU HI
            ;2889 ; EXPECTED OUTPUT OF ASHF AND FALU LO
            ;2890 ; CONTENTS OF LA DURING SHIFT <07:00>
            ;2891 ; CONTENTS OF LB DURING SHIFT <07:00>
            ;2892 ; EXPONENT DIFFERENCE (LB-LA) IN BITS <07:00>
            ;2893 ;
            ;2894 ; SYNC POINT DESCRIPTION
            ;2895 ;
            ;2896 ;--
            ;2897 ;*****************************************************************
            ;2898 =0
U 10B2, 0000.003D.0180.0800.0000.1138  ;2899 ASHF2: NEWTST
U 10B3, 0818.0038.8580.0800.0000.12DB  ;2900  D_K[.C]
U 12DB, 0019.0000.0580.09F8.0000.12DC  ;2901  ALU_D-K[.1],RC[0F]_ALU
U 12DC, 0018.0038.41F8.0AC0.0000.12DD  ;2902  R[8]_K[.80],Q_0  ; SET UP SOME CONSTANTS.
U 12DD, 0000.003C.03B0.0800.0000.12DE  ;2903  Q_Q.RIGHT.SI/1
U 12DE, 0000.003C.01B0.0800.0000.12DF  ;2904  Q_Q.RIGHT.SI/ZERO
U 12DF, 0001.203C.0180.09D0.0000.12E0  ;2905  RC[0A]_Q
U 12E0, 0003.003C.0180.09C8.0000.12E1  ;2906  RC[9]_0
U 12E1, 0018.0038.0580.09B0.0000.12E2  ;2907  RC[6]_K[.1]
U 12E2, 0018.0038.5580.0A90.0000.10B4  ;2908  R[2]_K[.3F]  ; INITIALIZE A COUNTER
U 10B4, 0000.003D.0180.0800.0000.114A  ;2909 =0 ERLOOP
U 10B5, 0000.003C.0180.0800.0000.10B6  ;2910  NOP
            ;2911 =0
U 10B6, 0000.003D.0180.0800.0000.1154  ;2912 ASHF2A: CALL,J/FPINIT
U 10B7, 0018.0038.2180.0800.0200.10B8  ;2913  VA_K[.14]  ; LOAD AN ADDD INSTRUCTION INTO
U 10B8, 0000.003D.0180.0800.0000.1160  ;2914 =0 CALL,J/LOADIB  ; THE IB.
U 10B9, 0818.0038.3180.0800.0000.10BA  ;2915  D_K[.40]  ; GENERATE THE ADDRESS OF THE
```

```
U 10BA. 0819.0031.D580.0800.0000.115A  ;2916 =0 ALU_D[OR]K[.6],D_ALU,CALL,J/GENTRA ; FPA UCODE USED.
U 10BB. 0000.003C.5980.3C00.0000.12E3  ;2917  ID[ACC.2]_D
U 12E3. 0000.00FC.0380.0800.0000.12E4  ;2918  TRAP.FPA  ; TRAP TO FPA LOCATION 46
U 12E4. 0000.003C.0180.0A40.0000.12E5  ;2919  LAB_R[8]  ; AT FPA 46
U 12E5. 0000.007C.0180.0A40.0000.12E6  ;2920  LAB_R[8],CPSYNC  ; AT FPA 46
U 12E6. 0000.003C.5DC0.0A10.0084.72E7  ;2921  Q_R[2],SC_K[.7]  ; COMPUTE LA AND LB VALUES
U 12E7. 0818.0038.5580.0800.0000.12E8  ;2922  D_K[.3F]  ; WHICH WILL RESULT IN THE
U 12E8. 081D.0000.0180.09A0.0000.12E9  ;2923  ALU_D-Q,D_ALU,RC[04]_ALU ; APPROPRIATE SHIFT COUNT.
U 12E9. 0819.0014.05F8.09A8.0000.12EA  ;2924  ALU_D+K[.1],D_ALU,RC[5]_ALU,Q_0
U 12EA. 0D00.003C.0180.0800.0000.12EB  ;2925  D_DAL.SC
U 12EB. 0001.003C.0180.0AD8.0000.12EC  ;2926  R[0B]_D
U 12EC. 0818.0038.3180.0800.0000.10BC  ;2927  D_K[.40]  ; GENERATE THE ADDRESS OF THE
U 10BC. 0819.0015.0980.0800.0000.115A  ;2928 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 10BD. 0000.003C.5980.3C00.0000.12ED  ;2929  ID[ACC.2]_D
U 12ED. 0000.00FC.0380.0800.0000.12EE  ;2930  TRAP.FPA  ; TRAP TO FPA 42
U 12EE. 0000.003C.0180.0A40.0000.12EF  ;2931  LAB_R[8]  ; AT FPA 42
U 12EF. 0000.007C.0180.0A40.0000.12F0  ;2932  LAB_R[8],CPSYNC  ; AT FPA 42
U 12F0. 0000.003C.0180.0A58.0000.12F1  ;2933  LAB_R[0B]  ; AT FPA 43
U 12F1. 0000.007C.0180.0A58.0000.12F2  ;2934  LAB_R[0B],CPSYNC ; AT FPA 43
U 12F2. 0000.003C.01D8.0800.0000.12F3  ;2935  Q_ACC  ; AT FPA 44, GET RESULT HI
U 12F3. 0A00.003C.0180.0800.0000.12F4  ;2936  D_ACC  ; AT FPA 45, GET RESULT LO
U 12F4. 0001.203C.0180.09E0.0000.12F5  ;2937  RC[0C]_Q
U 12F5. 0001.003C.0180.09D8.0000.12F6  ;2938  RC[0B]_D
U 12F6. 0010.0038.0180.0920.0082.12F7  ;2939  SC_RC[4]  ; COMPUTE THE EXPECTED RESULT.
U 12F7. 0F10.0038.01C0.0950.0000.12F8  ;2940  Q_RC[0A],D_0
U 12F8. 0000.003C.0180.0800.0010.12F9  ;2941 ASHF2B: EALU_SC,CLK.UBCC
U 12F9. 0000.123C.0180.0800.0000.104B  ;2942  EALU.Z?
U 104B. 0600.003C.04B0.0800.0084.B2F8  ;2943 =1011 Q_Q.RIGHT,D_D.RIGHT,SI/ASHR,SC_SC-K[.1],J/ASHF2B
U 104F. 0001.203C.0180.09C0.0000.12FA  ;2944  RC[8]_Q
U 12FA. 0001.003C.0180.09B8.0000.12FB  ;2945  RC[7]_D
U 12FB. 0001.203C.0180.0AA0.0000.12FC  ;2946  R[4]_Q
U 12FC. 0001.003C.0180.0AA8.0000.10BE  ;2947  R[5]_D
U 10BE. 0000.003D.0180.0800.0000.1182  ;2948 =0 CALL,J/UNSEQA
U 10BF. 0800.003C.0180.0A30.0000.12FD  ;2949  D_R[6]
U 12FD. 0000.003C.01C0.0A38.0000.12FE  ;2950  Q_R[7]
U 12FE. 0001.003C.0180.09F0.0000.12FF  ;2951  RC[0E]_D
U 12FF. 0001.203C.0180.09E8.0000.1300  ;2952  RC[0D]_Q
U 1300. 0010.0038.01C0.0960.0000.1301  ;2953  Q_RC[0C]
U 1301. 001D.0000.0180.0800.0010.1302  ;2954  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 1302. 0000.013C.0180.0800.0000.10C0  ;2955  Z?
U 10C0. 0000.003D.0180.0800.0000.114B  ;2956 =0 ERROR1  ; RESULT HI INCORRECT.
U 10C1. 0810.0038.0180.0968.0000.1303  ;2957  D_RC[0D]
U 1303. 0010.0038.01C0.0958.0000.1304  ;2958  Q_RC[0B]
U 1304. 001D.0000.0180.0800.0010.1305  ;2959  ALU_D-Q,CLK.UBCC
U 1305. 0000.013C.0180.0800.0000.10C2  ;2960  Z?
U 10C2. 0000.003D.0180.0800.0000.114B  ;2961 =0 ERROR1  ; RESULT LO INCORRECT.
U 10C3. 0000.003C.0180.0A10.0010.1306  ;2962  ALU_R[2],CLK.UBCC
U 1306. 0000.013C.0180.0800.0000.10C4  ;2963  Z?
U 10C4. 0018.0000.0580.0A90.0000.10B6  ;2964 =0 R[2]_LA-K[.1],J/ASHF2A
U 10C5. 0000.003C.0180.0800.0000.10C6  ;2965  NOP
```

```
;2966  .PAGE "TEST 268 ASHF AND EXPONENT DIFFERENCE TEST 3"
;2967 ;*****************************************************************
;2968 ;++
;2969 ; ASHF AND EXPONENT DIFFERENCE TEST 3
;2970 ;
;2971 ; TEST DESCRIPTION
;2972 ; THIS TEST SETS BIT 62 IN AR ALONG WITH SETTING ONE LOWER
;2973 ; ORDER BIT (GENERATED BY FLOATING A 1 ACROSS 0'S) IN AR.
;2974 ; THEN A FALU A+B IS DONE WITH EXPONENT COMBINATIONS WHICH
;2975 ; RESULT IN AR BEING SHIFTED RIGHT BY 63 THRU 0 PLACES.
;2976 ;
;2977 ; LOGIC DESCRIPTION
;2978 ;
;2979 ; FPA UCODE USED
;2980 ; LOCATION CONTENTS
;2981 ; 47:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/BR,WAIT,NAD/48
;2982 ; 48:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/49
;2983 ; 49:  BSC/R,SCR/CPU,FADC/ARO,WAIT,NAD/PSTALL
;2984 ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
;2985 ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
;2986 ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
;2987 ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
;2988 ;
;2989 ; ERROR DESCRIPTION
;2990 ; EXPECTED RESULT, FALU HI ON BUS A
;2991 ; EXPECTED RESULT, FALU LO ON BUS A
;2992 ; GOT RESULT HI
;2993 ; GOT RESULT LO
;2994 ; CONTENTS OF AR DURING SHIFT HI
;2995 ; CONTENTS OF AR DURING SHIFT LO
;2996 ; EXPECTED OUTPUT OF ASHF AND FALU HI
;2997 ; EXPECTED OUTPUT OF ASHF AND FALU LO
;2998 ; CONTENTS OF LA DURING SHIFT <07:00>
;2999 ; CONTENTS OF LB DURING SHIFT <07:00>
;3000 ; EXPONENT DIFFERENCE (LB-LA) IN BITS <07:00>
;3001 ;
;3002 ; SYNC POINT DESCRIPTION
;3003 ;
;3004 ;--
;3005 ;*****************************************************************
;3006 =0
U 10C6, 0000.003D.0180.0800.0000.1138  ;3007 ASHF3: NEWTST
U 10C7, 081B.0038.8580.0800.0000.1307  ;3008  D_K[.C]
U 1307, 0019.0000.0580.09F8.0000.1308  ;3009  ALU_D-K[.1],RC[0F]_ALU
U 1308, 001B.0038.4180.0AC0.0000.1309  ;3010  R[8]_K[.80]
U 1309, 001B.0038.05C0.0800.0000.130A  ;3011  Q_K[.1]
U 130A, 0000.003C.1980.0800.0084.730B  ;3012  SC_K[ZERO]
U 130B, 0F00.003C.0D80.0800.0084.B30C  ;3013  SC_SC-K[.3],D_0
U 130C, 0D00.003C.0180.0800.0000.130D  ;3014  D_DAL.SC
U 130D, 0001.003C.0180.0980.0000.130E  ;3015  RC[0]_D
U 130E, 0003.003C.05F8.0988.0084.730F  ;3016  RC[1]_0,Q_0,SC_K[.1]
U 130F, 0D00.003C.0180.0800.0000.1310  ;3017  D_DAL.SC
U 1310, 0001.003C.0180.0990.0000.1311  ;3018  RC[2]_D
U 1311, 001B.0038.0580.09B0.0000.1312  ;3019  RC[6]_K[.1]
U 1312, 081B.0038.5580.0800.0000.1313  ;3020  D_K[.3F]
```

```
U 1313. 0019.0000.0980.0A88.0000.1314  ;3021  ALU_D-K[.2],R[1]_ALU
U 1314. 0018.0038.5580.0A90.0000.10C8  ;3022  R[2]_K[.3F]
U 10C8. 0000.003D.0180.0800.0000.114A  ;3023 =0 ERLOOP
U 10C9. 0000.003C.0180.0800.0000.10CA  ;3024  NOP
        ;3025 =0
U 10CA. 0000.003D.0180.0800.0000.1154  ;3026 ASHF3A: CALL,J/FPINIT
U 10CB. 0018.0038.2180.0800.0200.10CC  ;3027  VA_K[.14]
U 10CC. 0000.003D.0180.0800.0000.1160  ;3028 =0 CALL,J/LOADIB  ; LOAD ADD INSTRUCTION IN IB.
U 10CD. 0810.0038.0180.0910.0000.1315  ;3029  D_RC[2]   ; GENERATE THE AR OPERAND
U 1315. 0010.0038.01C0.0900.0000.1316  ;3030  Q_RC[0]
U 1316. 081D.0030.0180.09D0.0000.1317  ;3031  ALU_D[OR]Q,D_ALU,RC[0A]_ALU
U 1317. 0001.003C.0180.0AA0.0000.1318  ;3032  R[4]_D
U 1318. 0810.0038.0180.0908.0000.1319  ;3033  D_RC[1]
U 1319. 0001.003C.0180.09C8.0000.10CE  ;3034  RC[9]_D
U 10CE. 0001.003D.0180.0AA8.0000.1182  ;3035 =0 R[5]_D,CALL,J/UNSEQA
U 10CF. 0818.0038.3180.0800.0000.10D0  ;3036  D_K[.40]  ; GENERATE THE ADDRESS OF
U 10D0. 0819.0031.5D80.0800.0000.115A  ;3037 =0 ALU_D[OR]K[.7],D_ALU,CALL,J/GENTRA ; THE FPA UCODE
U 10D1. 0000.003C.5980.3C00.0000.131A  ;3038  ID[ACC.2]_D
U 131A. 0000.00FC.0380.0800.0000.131B  ;3039  TRAP.FPA  ; TRAP TO FPA 47
U 131B. 0000.003C.0180.0A40.0000.131C  ;3040  LAB_R[8]  ; AT FPA 47
U 131C. 0000.007C.0180.0A40.0000.131D  ;3041  LAB_R[8],CPSYNC  ; AT FPA 47
U 131D. 0000.003C.0180.0A30.0000.131E  ;3042  LAB_R[6]  ; AT FPA 48
U 131E. 0000.007C.0180.0A30.0000.131F  ;3043  LAB_R[6],CPSYNC  ; AT FPA 48
U 131F. 0000.003C.0180.0A38.0000.1320  ;3044  LAB_R[7]  ; AT FPA 49
U 1320. 0000.007C.0180.0A38.0000.1321  ;3045  LAB_R[7],CPSYNC  ; AT FPA 49
U 1321. 0000.003C.5DC0.0A10.0084.7322  ;3046  Q_R[2],SC_K[.7]  ; GENERATE LA AND LB VALUES
U 1322. 0818.0038.5580.0800.0000.1323  ;3047  D_K[.3F]  ; WITH THE APPROPRIATE SHIFT COUNT.
U 1323. 081D.0000.0180.09A0.0000.1324  ;3048  ALU_D-Q,D_ALU,RC[4]_ALU
U 1324. 0819.0014.05F8.09A8.0000.1325  ;3049  ALU_D+K[.1],D_ALU,RC[5]_ALU,Q_0
U 1325. 0D00.003C.0180.0800.0000.1326  ;3050  D_DAL.SC
U 1326. 0001.003C.0180.0AD8.0000.1327  ;3051  R[0B]_D
U 1327. 0818.0038.3180.0800.0000.10D2  ;3052  D_K[.40]  ; GENERATE THE ADDRESS OF THE FPA UCODE.
U 10D2. 0819.0015.0980.0800.0000.115A  ;3053 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA
U 10D3. 0000.003C.5980.3C00.0000.1328  ;3054  ID[ACC.2]_D
U 1328. 0000.00FC.0380.0800.0000.1329  ;3055  TRAP.FPA  ; TRAP TO FPA 42
U 1329. 0000.003C.0180.0A40.0000.132A  ;3056  LAB_R[8]  ; AT FPA 42
U 132A. 0000.007C.0180.0A40.0000.132B  ;3057  LAB_R[8],CPSYNC  ; AT FPA 42
U 132B. 0000.003C.0180.0A58.0000.132C  ;3058  LAB_R[0B]  ; AT FPA 43
U 132C. 0000.007C.0180.0A58.0000.132D  ;3059  LAB_R[0B],CPSYNC ; AT FPA 43
U 132D. 0000.003C.01D8.0800.0000.132E  ;3060  Q_ACC   ; AT FPA 44, GET RESULT HI
U 132E. 0A00.003C.0180.0800.0000.132F  ;3061  D_ACC   ; AT FPA 45, GET RESULT LO
U 132F. 0001.203C.0180.09E0.0000.1330  ;3062  RC[0C]_Q
U 1330. 0001.003C.0180.09D8.0000.1331  ;3063  RC[0B]_D
U 1331. 0010.0038.0180.0920.0082.1332  ;3064  SC_RC[4]   ; COMPUTE THE EXPECTED RESULT.
U 1332. 0010.0038.01C0.0950.0000.1333  ;3065  Q_RC[0A]
U 1333. 0810.0038.0180.0948.0000.1334  ;3066  D_RC[9]
U 1334. 0000.003C.0180.0800.0010.1335  ;3067 ASHF3Z: EALU_SC,CLK.UBCC
U 1335. 0000.123C.0180.0800.0000.105B  ;3068  EALU.Z?
U 105B. 0600.003C.04B0.0800.0084.B334  ;3069 =1011 Q_Q.RIGHT,D_D.RIGHT,SI/ASHR,SC_SC-K[.1],J/ASHF3Z
U 105F. 0001.203C.0180.09C0.0000.1336  ;3070  RC[8]_Q
U 1336. 0001.003C.0180.09B8.0000.1337  ;3071  RC[7]_D
U 1337. 0001.203C.0180.0AA0.0000.1338  ;3072  R[4]_Q
U 1338. 0001.003C.0180.0AA8.0000.10D4  ;3073  R[5]_D
U 10D4. 0000.003D.0180.0800.0000.1182  ;3074 =0 CALL,J/UNSEQA
U 10D5. 0800.003C.0180.0A30.0000.1339  ;3075  D_R[6]
```

```
U 1339. 0000.003C.01C0.0A38.0000.133A  ;3076  Q_R[7]
U 133A. 0001.003C.0180.09F0.0000.133B  ;3077  RC[0E]_D
U 133B. 0001.203C.0180.09E8.0000.133C  ;3078  RC[0D]_Q
U 133C. 0010.0038.01C0.0960.0000.133D  ;3079  Q_RC[0C]
U 133D. 001D.0000.0180.0800.0010.133E  ;3080  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 133E. 0000.013C.0180.0800.0000.10D6  ;3081  Z?
U 10D6. 0000.003D.0180.0800.0000.114B  ;3082 =0 ERROR1    ; RESULT HI INCORRECT.
U 10D7. 0810.0038.0180.0968.0000.133F  ;3083  D_RC[0D]
U 133F. 0010.0038.01C0.0958.0000.1340  ;3084  Q_RC[0B]
U 1340. 001D.0000.0180.0800.0010.1341  ;3085  ALU_D-Q,CLK.UBCC
U 1341. 0000.013C.0180.0800.0000.10D8  ;3086  Z?
U 10D8. 0000.003D.0180.0800.0000.114B  ;3087 =0 ERROR1    ; RESULT LO INCORRECT.
U 10D9. 0000.003C.0180.0A10.0010.1342  ;3088  ALU_R[2],CLK.UBCC ; TRIED ALL SHIFT COUNTS?
U 1342. 0000.013C.0180.0800.0000.10DA  ;3089  Z?
U 10DA. 0018.0000.0580.0A90.0000.10CA  ;3090 =0 R[2]_LA-K[.1],J/ASHF3A ; IF NOT GET NEXT.
U 10DB. 0018.0038.5580.0A90.0000.1343  ;3091  R[2]_K[.3F]  ; IF YES THEN GET READY TO TEST NEXT AR PATTERN
U 1343. 0010.0038.01C0.0900.0000.1344  ;3092  Q_RC[0]    ; WITH ALL SHIFT COUNTS.
U 1344. 0810.0038.0180.0908.0000.1345  ;3093  D_RC[1]
U 1345. 0600.003C.00B0.0800.0000.1346  ;3094  D_D.RIGHT,Q_Q.RIGHT,SI/ASHR
U 1346. 0001.203C.0180.0980.0000.1347  ;3095  RC[0]_Q
U 1347. 0001.003C.0180.0988.0000.1348  ;3096  RC[1]_D
U 1348. 0000.003C.0180.0A08.0010.1349  ;3097  ALU_R[1],CLK.UBCC ; TRIED ALL AR PATTERNS?
U 1349. 0000.013C.0180.0800.0000.10DC  ;3098  Z?
U 10DC. 0018.0000.0580.0A88.0000.10CA  ;3099 =0 R[1]_LA-K[.1],J/ASHF3A ; IF NOT GET NEXT.
U 10DD. 0000.003C.0180.0800.0000.10DE  ;3100  NOP
```

```
                ;3101  .PAGE "TEST 269 ASHF AND EXPONENT DIFFERENCE TEST 4"
                ;3102  ;*****************************************************************
                ;3103  ;++
                ;3104  ; ASHF AND EXPONENT DIFFERENCE TEST 4
                ;3105  ;
                ;3106  ; TEST DESCRIPTION
                ;3107  ; THIS TEST SETS BIT 63 IN BR AND CLEARS AR AND THEN DOES A FALU
                ;3108  ; A+B WITH EXPONENT DIFFENCES OF 63 THRU 0.
                ;3109  ;
                ;3110  ; LOGIC DESCRIPTION
                ;3111  ;
                ;3112  ; FPA UCODE USED
                ;3113  ; LOCATION CONTENTS
                ;3114  ; 4A:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/LD,WAIT,NAD/4B
                ;3115  ; 4B:  FADC/R1,BSC/FAL.LH,NAD/PSTALL
                ;3116  ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
                ;3117  ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
                ;3118  ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
                ;3119  ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
                ;3120  ;
                ;3121  ; ERROR DESCRIPTION
                ;3122  ; EXPECTED RESULT, FALU HI ON BUS A
                ;3123  ; EXPECTED RESULT, FALU LO ON BUS A
                ;3124  ; GOT RESULT HI
                ;3125  ; GOT RESULT LO
                ;3126  ; CONTENTS OF BR DURING SHIFT HI
                ;3127  ; CONTENTS OF BR DURING SHIFT LO
                ;3128  ; EXPECTED OUTPUT OF ASHF AND FALU HI
                ;3129  ; EXPECTED OUTPUT OF ASHF AND FALU LO
                ;3130  ; CONTENTS OF LB DURING SHIFT <07:00>
                ;3131  ; CONTENTS OF LA DURING SHIFT <07:00>
                ;3132  ; EXPONENT DIFFERENCE (LA-LB) IN BITS <07:00>
                ;3133  ;
                ;3134  ; SYNC POINT DESCRIPTION
                ;3135  ;
                ;3136  ;--
                ;3137  ;*****************************************************************
                ;3138  =0
U 10DE, 0000.003D,0180.0800.0000.1138  ;3139 ASHG1: NEWTST
U 10DF, 0818.0038,8580.0800.0000.134A  ;3140  D_K[.C]
U 134A, 0019.0000,0580.09F8.0000.134B  ;3141  ALU_D-K[.1],RC[0F]_ALU
U 134B, 0018.0038,41F8.0AC0.0000.134C  ;3142  R[8]_K[.80],Q_0  ; INITIALIZE SOME CONSTANTS
U 134C, 0000.003C,03B0.0800.0000.134D  ;3143  Q_Q.RIGHT,SI/7
U 134D, 0001.203C,0180.09D0.0000.134E  ;3144  RC[0A]_Q
U 134E, 0003.003C,0180.09C8.0000.134F  ;3145  RC[9]_0
U 134F, 0018.0038,0580.09B0.0000.1350  ;3146  RC[6]_K[.1]
U 1350, 0018.0038,5580.0A90.0000.10E0  ;3147  R[2]_K[.3F]  ; INITIALIZE A COUNTER
U 10E0, 0000.003D,0180.0800.0000.114A  ;3148 =0 ERLOOP
U 10E1, 0000.003C,0180.0800.0000.10E2  ;3149  NOP
                ;3150  =0
U 10E2, 0000.003D,0180.0800.0000.1154  ;3151 ASHG1A: CALL,J/FPINIT
U 10E3, 0018.0038,2180.0800.0200.10E4  ;3152  VA_K[.14]  ; LOAD ADDD INSTRUCTIO
U 10E4, 0000.003D,0180.0800.0000.1160  ;3153 =0 CALL,J/LOADIB  ; INTO THE IB.
U 10E5, 0818.0038,3180.0800.0000.10E6  ;3154  D_K[.40]  ; GENERATE ADDRESS OF THE FPA
U 10E6, 0819.0031,F580.0800.0000.115A  ;3155 =0 ALU_D[OR]K[.A],D_ALU,CALL,J/GENTRA ; UCODE USED.
```

```
U 10E7. 0000.003C.5980.3C00.0000.1351  ;3156  ID[ACC.2]_D
U 1351. 0000.00FC.0380.0800.0000.1352  ;3157  TRAP.FPA  ; TRAP TO FPA 4A
U 1352. 0000.003C.0180.0A40.0000.1353  ;3158  LAB_R[8]  ; AT FPA 4A
U 1353. 0000.007C.0180.0A40.0000.1354  ;3159  LAB_R[8],CPSYNC  ; AT FPA 4A
U 1354. 0000.003C.0180.0800.0000.1355  ;3160  NOP  ; AT FPA 4B, BR BIT 63 SET AND
                ;3161     ; AR CLEARED HERE.
U 1355. 0000.003C.5DC0.0A10.0084.7356  ;3162  Q_R[2],SC_K[.7]  ; COMPUTE THE LA AND LB VALUES
U 1356. 0818.0038.5580.0800.0000.1357  ;3163  D_K[.3F]  ; APPROPRIATE FOR THIS SHIFT COUNT.
U 1357. 081D.0000.0180.09A0.0000.1358  ;3164  ALU_D-Q,D_ALU,RC[04]_ALU
U 1358. 0819.0014.05F8.09A8.0000.1359  ;3165  ALU_D+K[.1],D_ALU,RC[05]_ALU,Q_0
U 1359. 0D00.003C.0180.0800.0000.135A  ;3166  D_DAL.SC
U 135A. 0001.003C.0180.0AD8.0000.135B  ;3167  R[0B]_D
U 135B. 0818.0038.3180.0800.0000.10E8  ;3168  D_K[.40]
U 10E8. 0819.0015.0980.0800.0000.115A  ;3169 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; GENERATE ADDRESS OF FPA UCODE.
U 10E9. 0000.003C.5980.3C00.0000.135C  ;3170  ID[ACC.2]_D
U 135C. 0000.00FC.0380.0800.0000.135D  ;3171  TRAP.FPA  ; TRAP TO FPA 42
U 135D. 0000.003C.0180.0A58.0000.135E  ;3172  LAB_R[0B]  ; AT FPA 42
U 135E. 0000.007C.0180.0A58.0000.135F  ;3173  LAB_R[0B],CPSYNC  ; AT FPA 42
U 135F. 0000.003C.0180.0A40.0000.1360  ;3174  LAB_R[8]  ; AT FPA 43
U 1360. 0000.007C.0180.0A40.0000.1361  ;3175  LAB_R[8],CPSYNC  ; AT FPA 43
U 1361. 0000.003C.01D8.0800.0000.1362  ;3176  Q_ACC  ; AT FPA 44, GET RESULT HI
U 1362. 0A00.003C.0180.0800.0000.1363  ;3177  D_ACC  ; AT FPA 45, GET RESULT LO
U 1363. 0001.203C.0180.09E0.0000.1364  ;3178  RC[0C]_Q
U 1364. 0001.003C.0180.09D8.0000.1365  ;3179  RC[0B]_D
U 1365. 0000.003C.0180.0A10.0082.1366  ;3180  SC_R[2]  ; COMPUTE EXPECTED RESULT.
U 1366. 0810.0038.0180.0920.0000.1367  ;3181  D_RC[04]
U 1367. 0019.0034.7580.0800.0010.1368  ;3182  ALU_D[AND]K[.20],CLK.UBCC
U 1368. 0000.013C.0180.0800.0000.10EA  ;3183  Z?
U 10EA. 0000.003C.0180.0800.0000.136A  ;3184 =0 J/ASHG1B
U 10EB. 0803.0010.0180.0800.0000.1369  ;3185  ALU_MASK+1,D_ALU
U 1369. 0000.003C.01F8.0800.0000.136C  ;3186  Q_0,J/ASHG1C
U 136A. 081B.0000.0580.0800.0000.136B  ;3187 ASHG1B: ALU_0-K[.1],D_ALU
U 136B. 0003.0010.01C0.0800.0000.136C  ;3188  ALU_MASK+1,Q_ALU
U 136C. 0001.003C.0180.09C0.0000.136D  ;3189 ASHG1C: RC[0]_D
U 136D. 0001.203C.0180.09B8.0000.136E  ;3190  RC[7]_Q
U 136E. 0001.003C.0180.0AA0.0000.10EC  ;3191  R[4]_D
U 10EC. 0001.203D.0180.0AA8.0000.1182  ;3192 =0 R[5]_Q,CALL,J/UNSEQA
U 10ED. 0800.003C.0180.0A30.0000.136F  ;3193  D_R[6]
U 136F. 0000.003C.01C0.0A38.0000.1370  ;3194  Q_R[7]
U 1370. 0001.003C.0180.09F0.0000.1371  ;3195  RC[0E]_D
U 1371. 0001.203C.0180.09E8.0000.1372  ;3196  RC[0D]_Q
U 1372. 0010.0038.01C0.0960.0000.1373  ;3197  Q_RC[0C]
U 1373. 001D.0000.0180.0800.0010.1374  ;3198  ALU_D-Q,CLK.UBCC ; CHECK RESULT
U 1374. 0000.013C.0180.0800.0000.10EE  ;3199  Z?
U 10EE. 0000.003D.0180.0800.0000.114B  ;3200 =0 ERROR1  ; RESULT HI INCORRECT.
U 10EF. 0810.0038.0180.0968.0000.1375  ;3201  D_RC[0D]
U 1375. 0010.0038.01C0.0958.0000.1376  ;3202  Q_RC[0B]
U 1376. 001D.0000.0180.0800.0010.1377  ;3203  ALU_D-Q,CLK.UBCC
U 1377. 0000.013C.0180.0800.0000.10F0  ;3204  Z?
U 10F0. 0000.003D.0180.0800.0000.114B  ;3205 =0 ERROR1  ; RESULT LO INCORRECT.
U 10F1. 0000.003C.0180.0A10.0010.1378  ;3206  ALU_R[2],CLK.UBCC ; TRIED ALL SHIFT COUNTS?
U 1378. 0000.013C.0180.0800.0000.10F2  ;3207  Z?
U 10F2. 0018.0000.0580.0A90.0000.10E2  ;3208 =0 R[2]_LA-K[.1],J/ASHG1A ; NO SO CONTINUE TEST.
U 10F3. 0000.003C.0180.0800.0000.10F4  ;3209  NOP
```

```
                ;3210  .PAGE "TEST 26A ASHF AND EXPONENT DIFFERENCE TEST 5"
                ;3211  ;*****************************************************************
                ;3212  ;++
                ;3213  ; ASHF AND EXPONENT DIFFERENCE TEST 5
                ;3214  ;
                ;3215  ; TEST DESCRIPTION
                ;3216  ; THIS TEST SETS JUST BIT 62 IN THE AR AND BR AND THEN
                ;3217  ; DOES A FALU A+B FUNCTION WITH EXPONENTS SUCH THAT
                ;3218  ; BR IS SHIFTED RIGHT WITH A SHIFT COUNT OF 63 THRU 0.
                ;3219  ;
                ;3220  ; LOGIC DESCRIPTION
                ;3221  ;
                ;3222  ; FPA UCODE USED
                ;3223  ; LOCATION CONTENTS
                ;3224  ; 46:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/LD,WAIT,NAD/PSTALL
                ;3225  ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
                ;3226  ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
                ;3227  ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
                ;3228  ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
                ;3229  ;
                ;3230  ; ERROR DESCRIPTION
                ;3231  ; EXPECTED RESULT, FALU HI ON BUS A
                ;3232  ; EXPECTED RESULT, FALU LO ON BUS A
                ;3233  ; GOT RESULT HI
                ;3234  ; GOT RESULT LO
                ;3235  ; CONTENTS OF BR DURING SHIFT HI
                ;3236  ; CONTENTS OF BR DURING SHIFT LO
                ;3237  ; EXPECTED OUTPUT OF ASHF AND FALU HI
                ;3238  ; EXPECTED OUTPUT OF ASHF AND FALU LO
                ;3239  ; CONTENTS OF LB DURING SHIFT <07:00>
                ;3240  ; CONTENTS OF LA DURING SHIFT <07:00>
                ;3241  ; EXPONENT DIFFERENCE (LA-LB) IN BITS <07:00>
                ;3242  ;
                ;3243  ; SYNC POINT DESCRIPTION
                ;3244  ;
                ;3245  ;--
                ;3246  ;*****************************************************************
                ;3247  =0
U 10F4, 0000.003D.0180.0800.0000.1138  ;3248 ASHG2: NEWTST
U 10F5, 0818.0038.8580.0800.0000.1379  ;3249  D_K[.C]
U 1379, 0019.0000.0580.09F8.0000.137A  ;3250  ALU_D-K[.1],RC[0F]_ALU
U 137A, 0018.0038.41F8.0AC0.0000.137B  ;3251  R[8]_K[.80],Q_0  ; SET UP SOME CONSTANTS.
U 137B, 0000.003C.03B0.0800.0000.137C  ;3252  Q_Q.RIGHT,SI/1
U 137C, 0000.003C.01B0.0800.0000.137D  ;3253  Q_Q.RIGHT,SI/ZERO
U 137D, 0001.203C.0180.09D0.0000.137E  ;3254  RC[0A]_Q
U 137E, 0003.003C.0180.09C8.0000.137F  ;3255  RC[9]_0
U 137F, 0018.0038.0580.09B0.0000.1380  ;3256  RC[6]_K[.1]
U 1380, 0018.0038.5580.0A90.0000.10F6  ;3257  R[2]_K[.3F]  ; INITIALIZE A COUNTER
U 10F6, 0000.003D.0180.0800.0000.114A  ;3258 =0 ERLOOP
U 10F7, 0000.003C.0180.0800.0000.10F8  ;3259  NOP
                ;3260  =0
U 10F8, 0000.003D.0180.0800.0000.1154  ;3261 ASHG2A: CALL,J/FPINIT
U 10F9, 0018.0038.2180.0800.0200.10FA  ;3262  VA_K[.14]  ; LOAD AN ADDD INSTRUCTION INTO
U 10FA, 0000.003D.0180.0800.0000.1160  ;3263 =0 CALL,J/LOADIB  ; THE IB.
U 10FB, 0818.0038.3180.0800.0000.10FC  ;3264  D_K[.40]  ; GENERATE THE ADDRESS OF THE
```

```
U 10FC. 0819,0031,D580,0800,0000,115A  ;3265 =0 ALU_D[OR]K[.6],D_ALU,CALL,J/GENTRA ; FPA UCODE USED.
U 10FD. 0000,003C,5980,3C00,0000,1381  ;3266  ID[ACC.2]_D
U 1381. 0000,00FC,0380,0800,0000,1382  ;3267  TRAP.FPA  ; TRAP TO FPA LOCATION 46
U 1382. 0000,003C,0180,0A40,0000,1383  ;3268  LAB_R[8]  ; AT FPA 46
U 1383. 0000,007C,0180,0A40,0000,1384  ;3269  LAB_R[8],CPSYNC  ; AT FPA 46
U 1384. 0000,003C,5DC0,0A10,0084,7385  ;3270  Q_R[2],SC_K[.7]  ; COMPUTE LA AND LB VALUES
U 1385. 0818,0038,5580,0800,0000,1386  ;3271  D_K[.3F]  ; WHICH WILL RESULT IN THE
U 1386. 081D,0000,0180,09A0,0000,1387  ;3272  ALU_D-Q,D_ALU,RC[04]_ALU ; APPROPRIATE SHIFT COUNT.
U 1387. 0819,0014,05F8,09A8,0000,1388  ;3273  ALU_D+K[.1],D_ALU,RC[5]_ALU,Q_0
U 1388. 0D00,003C,0180,0800,0000,1389  ;3274  D_DAL.SC
U 1389. 0001,003C,0180,0AD8,0000,138A  ;3275  R[0B]_D
U 138A. 0818,0038,3180,0800,0000,10FE  ;3276  D_K[.40]  ; GENERATE THE ADDRESS OF THE
U 10FE. 0819,0015,0980,0800,0000,115A  ;3277 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA ; FPA UCODE.
U 10FF. 0000,003C,5980,3C00,0000,138B  ;3278  ID[ACC.2]_D
U 138B. 0000,00FC,0380,0800,0000,138C  ;3279  TRAP.FPA  ; TRAP TO FPA 42
U 138C. 0000,003C,0180,0A58,0000,138D  ;3280  LAB_R[0B]  ; AT FPA 42
U 138D. 0000,007C,0180,0A58,0000,138E  ;3281  LAB_R[0B],CPSYNC  ; AT FPA 42
U 138E. 0000,003C,0180,0A40,0000,138F  ;3282  LAB_R[8]  ; AT FPA 43
U 138F. 0000,007C,0180,0A40,0000,1390  ;3283  LAB_R[8],CPSYNC  ; AT FPA 43
U 1390. 0000,003C,01D8,0800,0000,1391  ;3284  Q_ACC  ; AT FPA 44, GET RESULT HI
U 1391. 0A00,003C,0180,0800,0000,1392  ;3285  D_ACC  ; AT FPA 45, GET RESULT LO
U 1392. 0001,203C,0180,09E0,0000,1393  ;3286  RC[0C]_Q
U 1393. 0001,003C,0180,09D8,0000,1394  ;3287  RC[0B]_D
U 1394. 0010,0038,0180,0920,0082,1395  ;3288  SC_RC[4]  ; COMPUTE THE EXPECTED RESULT.
U 1395. 0F10,0038,01C0,0950,0000,1396  ;3289  Q_RC[0A],D_0
U 1396. 0000,003C,0180,0800,0010,1397  ;3290 ASHG2B: EALU_SC,CLK.UBCC
U 1397. 0000,123C,0180,0800,0000,106B  ;3291  EALU.Z?
U 106B. 0600,003C,04B0,0800,0084,B396  ;3292 =1011 Q_Q.RIGHT,D_D.RIGHT,SI/ASHR,SC_SC-K[.1],J/ASHG2B
U 106F. 0001,203C,0180,09C0,0000,1398  ;3293  RC[8]_Q
U 1398. 0001,003C,0180,09B8,0000,1399  ;3294  RC[7]_D
U 1399. 0001,203C,0180,0AA0,0000,139A  ;3295  R[4]_Q
U 139A. 0001,003C,0180,0AA8,0000,110A  ;3296  R[5]_D
U 110A. 0000,003D,0180,0800,0000,1182  ;3297 =0 CALL,J/UNSEQA
U 110B. 0800,003C,0180,0A30,0000,139B  ;3298  D_R[6]
U 139B. 0000,003C,01C0,0A38,0000,139C  ;3299  Q_R[7]
U 139C. 0001,003C,0180,09F0,0000,139D  ;3300  RC[0E]_D
U 139D. 0001,203C,0180,09E8,0000,139E  ;3301  RC[0D]_Q
U 139E. 0010,0038,01C0,0960,0000,139F  ;3302  Q_RC[0C]
U 139F. 001D,0000,0180,0800,0010,13A0  ;3303  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 13A0. 0000,013C,0180,0800,0000,1110  ;3304  Z?
U 1110. 0000,003D,0180,0800,0000,114B  ;3305 =0 ERROR1  ; RESULT HI INCORRECT.
U 1111. 0810,0038,0180,0968,0000,13A1  ;3306  D_RC[0D]
U 13A1. 0010,0038,01C0,0958,0000,13A2  ;3307  Q_RC[0B]
U 13A2. 001D,0000,0180,0800,0010,13A3  ;3308  ALU_D-Q,CLK.UBCC
U 13A3. 0000,013C,0180,0800,0000,1112  ;3309  Z?
U 1112. 0000,003D,0180,0800,0000,114B  ;3310 =0 ERROR1  ; RESULT LO INCORRECT.
U 1113. 0000,003C,0180,0A10,0010,13A4  ;3311  ALU_R[2],CLK.UBCC
U 13A4. 0000,013C,0180,0800,0000,1114  ;3312  Z?
U 1114. 0018,0000,0580,0A90,0000,10F8  ;3313 =0 R[2]_LA-K[.1],J/ASHG2A
U 1115. 0000,003C,0180,0800,0000,1116  ;3314  NOP
```

```
;3315  .PAGE "TEST 26B ASHF AND EXPONENT DIFFERENCE TEST 6"
;3316 ;*************************************************************
;3317 ;**
;3318 ; ASHF AND EXPONENT DIFFERENCE TEST 6
;3319 ;
;3320 ; TEST DESCRIPTION
;3321 ; THIS TEST SETS BIT 62 IN BR ALONG WITH SETTING ONE LOWER
;3322 ; ORDER BIT (GENERATED BY FLOATING A 1 ACROSS 0'S) IN BR.
;3323 ; THEN A FALU A+B IS DONE WITH EXPONENT COMBINATIONS WHICH
;3324 ; RESULTS IN BR BEING SHIFTED RIGHT BY 63 THRU 0 PLACES.
;3325 ;
;3326 ; LOGIC DESCRIPTION
;3327 ;
;3328 ; FPA UCODE USED
;3329 ; LOCATION CONTENTS
;3330 ; 4C:  BSC/R,SCR/CPU,EAC/LDAB,SGNC/LDS,FADC/AR,WAIT,NAD/4D
;3331 ; 4D:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/4E
;3332 ; 4E:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/PSTALL
;3333 ; 42:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/43
;3334 ; 43:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/44
;3335 ; 44:  FADC/A.B,BSC/FAL.HL,NAD/45
;3336 ; 45:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
;3337 ;
;3338 ; ERROR DESCRIPTION
;3339 ; EXPECTED RESULT, FALU HI ON BUS A
;3340 ; EXPECTED RESULT, FALU LO ON BUS A
;3341 ; GOT RESULT HI
;3342 ; GOT RESULT LO
;3343 ; CONTENTS OF BR DURING SHIFT HI
;3344 ; CONTENTS OF BR DURING SHIFT LO
;3345 ; EXPECTED OUTPUT OF ASHF AND FALU HI
;3346 ; EXPECTED OUTPUT OF ASHF AND FALU LO
;3347 ; CONTENTS OF LB DURING SHIFT <07:00>
;3348 ; CONTENTS OF LA DURING SHIFT <07:00>
;3349 ; EXPONENT DIFFERENCE (LA-LB) IN BITS <07:00>
;3350 ;
;3351 ; SYNC POINT DESCRIPTION
;3352 ;
;3353 ;--
;3354 ;*************************************************************
;3355 =0
U 1116, 0000.003D.0180.0800.0000.1138  ;3356 ASHG3: NEWTST
U 1117, 0818.0038.8580.0800.0000.13A5  ;3357  D_K[.C]
U 13A5, 0019.0000.0580.09F8.0000.13A6  ;3358  ALU_D-K[.1],RC[0F]_ALU
U 13A6, 0018.0038.4180.0AC0.0000.13A7  ;3359  R[8]_K[.80]
U 13A7, 0018.0038.05C0.0800.0000.13A8  ;3360  Q_K[.1]
U 13A8, 0000.003C.1980.0800.0084.73A9  ;3361  SC_K[ZERO]
U 13A9, 0F00.003C.0D80.0800.0084.B3AA  ;3362  SC_SC-K[.3],D_0
U 13AA, 0D00.003C.0180.0800.0000.13AB  ;3363  D_DAL.SC
U 13AB, 0001.003C.0180.0980.0000.13AC  ;3364  RC[0]_D
U 13AC, 0003.003C.05F8.0988.0084.73AD  ;3365  RC[1]_0,Q_0,SC_K[.1]
U 13AD, 0D00.003C.0180.0800.0000.13AE  ;3366  D_DAL.SC
U 13AE, 0001.003C.0180.0990.0000.13AF  ;3367  RC[2]_D
U 13AF, 0018.0038.0580.09B0.0000.13B0  ;3368  RC[6]_K[.1]
U 13B0, 0818.0038.2980.0800.0000.13B1  ;3369  D_K[.34]
```

```
U 13B1. 0019.0000.0980.0A88.0000.13B2  ;3370  ALU_D-K[.2],R[1]_ALU
U 13B2. 0018.0038.5580.0A90.0000.1118  ;3371  R[2]_K[.3F]
U 1118. 0000.003D.0180.0800.0000.114A  ;3372 =0 ERLOOP
U 1119. 0000.003C.0180.0800.0000.111A  ;3373  NOP
        ;3374 =0
U 111A. 0000.003D.0180.0800.0000.1154  ;3375 ASHG3A: CALL,J/FPINIT
U 111B. 0018.0038.2180.0800.0200.111C  ;3376  VA_K[.14]
U 111C. 0000.003D.0180.0800.0000.1160  ;3377 =0 CALL,J/LOADIB  ; LOAD ADD INSTRUCTION IN IB.
U 111D. 0810.0038.0180.0910.0000.13B3  ;3378  D_RC[2]   ; GENERATE THE AR OPERAND
U 13B3. 0010.0038.01C0.0900.0000.13B4  ;3379  Q_RC[0]
U 13B4. 081D.0030.0180.09D0.0000.13B5  ;3380  ALU_D[OR]Q,D_ALU,RC[0A]_ALU
U 13B5. 0001.003C.0180.0AA0.0000.13B6  ;3381  R[4]_D
U 13B6. 0810.0038.0180.0908.0000.13B7  ;3382  D_RC[1]
U 13B7. 0001.003C.0180.09C8.0000.111E  ;3383  RC[9]_D
U 111E. 0001.003D.0180.0AA8.0000.1190  ;3384 =0 R[5]_D,CALL,J/UNSEQB
U 111F. 0818.0038.3180.0800.0000.1120  ;3385  D_K[.40]  ; GENERATE THE ADDRESS OF
U 1120. 0819.0031.8580.0800.0000.115A  ;3386 =0 ALU_D[OR]K[.C],D_ALU,CALL,J/GENTRA ; THE FPA UCODE
U 1121. 0000.003C.5980.3C00.0000.13B8  ;3387  ID[ACC.2]_D
U 13B8. 0000.00FC.0380.0800.0000.13B9  ;3388  TRAP,FPA  ; TRAP TO FPA 4C
U 13B9. 0000.003C.0180.0A40.0000.13BA  ;3389  LAB_R[8]  ; AT FPA 4C
U 13BA. 0000.007C.0180.0A40.0000.13BB  ;3390  LAB_R[8],CPSYNC  ; AT FPA 4C
U 13BB. 0000.003C.0180.0A30.0000.13BC  ;3391  LAB_R[6]  ; AT FPA 4D
U 13BC. 0000.007C.0180.0A30.0000.13BD  ;3392  LAB_R[6],CPSYNC  ; AT FPA 4D
U 13BD. 0000.003C.0180.0A38.0000.13BE  ;3393  LAB_R[7]  ; AT FPA 4E
U 13BE. 0000.007C.0180.0A38.0000.13BF  ;3394  LAB_R[7],CPSYNC  ; AT FPA 4E
U 13BF. 0000.003C.5DC0.0A10.0084.73C0  ;3395  Q_R[2],SC_K[.7]  ; GENERATE LA AND LB VALUES
U 13C0. 0818.0038.5580.0800.0000.13C1  ;3396  D_K[.3F]  ; WITH THE APPROPRIATE SHIFT COUNT.
U 13C1. 081D.0000.0180.09A0.0000.13C2  ;3397  ALU_D-Q,D_ALU,RC[4]_ALU
U 13C2. 0819.0014.05F8.09A8.0000.13C3  ;3398  ALU_D+K[.1],D_ALU,RC[5]_ALU,Q_0
U 13C3. 0D00.003C.0180.0800.0000.13C4  ;3399  D_DAL.SC
U 13C4. 0001.003C.0180.0AD8.0000.13C5  ;3400  R[0B]_D
U 13C5. 0818.0038.3180.0800.0000.1122  ;3401  D_K[.40]  ; GENERATE THE ADDRESS OF THE FPA UCODE.
U 1122. 0819.0015.0980.0800.0000.115A  ;3402 =0 ALU_D+K[.2],D_ALU,CALL,J/GENTRA
U 1123. 0000.003C.5980.3C00.0000.13C6  ;3403  ID[ACC.2]_D
U 13C6. 0000.00FC.0380.0800.0000.13C7  ;3404  TRAP,FPA  ; TRAP TO FPA 42
U 13C7. 0000.003C.0180.0A58.0000.13C8  ;3405  LAB_R[0B]  ; AT FPA 42
U 13C8. 0000.007C.0180.0A58.0000.13C9  ;3406  LAB_R[0B],CPSYNC  ; AT FPA 42
U 13C9. 0000.003C.0180.0A40.0000.13CA  ;3407  LAB_R[8]  ; AT FPA 43
U 13CA. 0000.007C.0180.0A40.0000.13CB  ;3408  LAB_R[8],CPSYNC  ; AT FPA 43
U 13CB. 0000.003C.01D8.0800.0000.13CC  ;3409  Q_ACC  ; AT FPA 44. GET RESULT HI
U 13CC. 0A00.003C.0180.0800.0000.13CD  ;3410  D_ACC  ; AT FPA 45. GET RESULT LO
U 13CD. 0001.203C.0180.09E0.0000.13CE  ;3411  RC[0C]_Q
U 13CE. 0001.003C.0180.09D8.0000.13CF  ;3412  RC[0B]_D
U 13CF. 0010.0038.0180.0920.0082.13D0  ;3413  SC_RC[4]  ; COMPUTE THE EXPECTED RESULT.
U 13D0. 0010.0038.01C0.0950.0000.13D1  ;3414  Q_RC[0A]
U 13D1. 0810.0038.0180.0948.0000.13D2  ;3415  D_RC[9]
U 13D2. 0000.003C.0180.0800.0010.13D3  ;3416 ASHG3Z: EALU_SC,CLK.UBCC
U 13D3. 0000.123C.0180.0800.0000.107B  ;3417  EALU.Z?
U 107B. 0600.003C.04B0.0800.0084.B3D2  ;3418 =1011 Q_Q.RIGHT,D_D.RIGHT,SI/ASHR,SC_SC-K[.1],J/ASHG3Z
U 107F. 0001.203C.0180.09C0.0000.13D4  ;3419  RC[8]_Q
U 13D4. 0001.003C.0180.09B8.0000.13D5  ;3420  RC[7]_D
U 13D5. 0001.203C.0180.0AA0.0000.13D6  ;3421  R[4]_Q
U 13D6. 0001.003C.0180.0AA8.0000.1124  ;3422  R[5]_D
U 1124. 0000.003D.0180.0800.0000.1182  ;3423 =0 CALL,J/UNSEQA
U 1125. 0800.003C.0180.0A30.0000.13D7  ;3424  D_R[6]
```

```
U 13D7. 0000,003C,01C0,0A38,0000,13D8  ;3425  Q_R[7]
U 13D8. 0001,003C,0180,09F0,0000,13D9  ;3426  RC[0E]_D
U 13D9. 0001,203C,0180,09E8,0000,13DA  ;3427  RC[0D]_Q
U 13DA. 0010,0038,01C0,0960,0000,13DB  ;3428  Q_RC[0C]
U 13DB. 001D,0000,0180,0800,0010,13DC  ;3429  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 13DC. 0000,013C,0180,0800,0000,1126  ;3430  Z?
U 1126. 0000,003D,0180,0800,0000,114B  ;3431 =0 ERROR1    ; RESULT HI INCORRECT.
U 1127. 0810,0038,0180,0968,0000,13DD  ;3432  D_RC[0D]
U 13DD. 0010,0038,01C0,0958,0000,13DE  ;3433  Q_RC[0B]
U 13DE. 001D,0000,0180,0800,0010,13DF  ;3434  ALU_D-Q,CLK.UBCC
U 13DF. 0000,013C,0180,0800,0000,1128  ;3435  Z?
U 1128. 0000,003D,0180,0800,0000,114B  ;3436 =0 ERROR1    ; RESULT LO INCORRECT.
U 1129. 0000,003C,0180,0A10,0010,13E0  ;3437  ALU_R[2],CLK.UBCC ; TRIED ALL SHIFT COUNTS?
U 13E0. 0000,013C,0180,0800,0000,112A  ;3438  Z?
U 112A. 0018,0000,0580,0A90,0000,111A  ;3439 =0 R[2]_LA-K[.1],J/ASHG3A ; IF NOT GET NEXT.
U 112B. 0018,0038,5580,0A90,0000,13E1  ;3440  R[2]_K[.3F]  ; IF YES THEN GET READY TO TEST NEXT BR PATTERN
U 13E1. 0010,0038,01C0,0900,0000,13E2  ;3441  Q_RC[0]    ; WITH ALL SHIFT COUNTS.
U 13E2. 0810,0038,0180,0908,0000,13E3  ;3442  D_RC[1]
U 13E3. 0600,003C,00B0,0800,0000,13E4  ;3443  D_D.RIGHT,Q_Q.RIGHT,SI/ASHR
U 13E4. 0001,203C,0180,0980,0000,13E5  ;3444  RC[0]_Q
U 13E5. 0019,0024,5980,0988,0000,13E6  ;3445  ALU_D[ANDNOT]K[.7F],RC[1]_ALU
U 13E6. 0000,003C,0180,0A08,0010,13E7  ;3446  ALU_R[1],CLK.UBCC ; TRIED ALL BR PATTERNS?
U 13E7. 0000,013C,0180,0800,0000,112C  ;3447  Z?
U 112C. 0018,0000,0580,0A88,0000,111A  ;3448 =0 R[1]_LA-K[.1],J/ASHG3A ; IF NOT GET NEXT.
U 112D. 0000,003C,0180,0800,0000,112E  ;3449  NOP
          ;3450
```

```
            ;3451  .PAGE "TEST 26C RESERVED"
            ;3452 =0
U 112E. 0000,003D,0180,0800,0000,1138  ;3453 D2: NEWTST
U 112F. 0000,003C,0180,0800,0000,1130  ;3454  NOP
            ;3455
```

```
          ;3456  .PAGE "DUMMY NEWTST"
          ;3457 =0
U 1130, 0000,003D,0180,0800,0000,1138  ;3458 D3: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2539
ESKAR5E.MCR                    MICRO2  1P(00)    26-JUL-85  17:02:40     Page   8
:                            Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1907: 1874 1881= 1897= 1898= 1899= 1918= 1919=
U 1008   1995= 1996= 1998= 1999= 2000= 2001= 2058= 2059=
U 1010   2003= 2004= 2023= 2024= 2172= 2173= 2186= 2187=
U 1018   2200= 2201= 2211= 2212= 2269= 2270= 2077= 2078=
U 1020   2279= 2280= 2282= 2283= 2292= 2294= 2313= 2314=
U 1028   2323= 2324= 2326= 2327= 2330= 2331= 2095= 2096=
U 1030   2336= 2338= 2379= 2380= 2389= 2390= 2392= 2393=
U 1038   2404= 2406= 2425= 2426= 2435= 2436= 2117= 2118=
U 1040   2438= 2439= 2442= 2443= 2448= 2450= 2506= 2507=
U 1048   2509= 2510= 1875 2943= 2512= 2513= 1939: 2944=
U 1050   2515= 2516= 2526= 2527= 2531= 2532= 2543= 2544=
U 1058   2545= 2546= 1942: 3069= 2557= 2558= 1945: 3070=
U 1060   2564= 2565= 2567= 2568= 2579= 2580= 2581= 2582=
U 1068   2593= 2594= 1876 3292= 2600= 2601= 1877 3293=
U 1070   2605= 2606= 2616= 2617= 2618= 2619= 2630= 2631=
U 1078   2637= 2638= 1878 3418= 2641= 2642= 1879 3419=
U 1080   2653= 2654= 2660= 2661= 2662= 2663= 2674= 2675=
U 1088   2681= 2682= 2683= 2684= 2695= 2696= 2702= 2703=
U 1090   2706= 2707= 2718= 2719= 2725= 2726= 2730= 2731=
U 1098   2742= 2744= 2746= 2747= 2786= 2787= 2795= 2796=
U 10A0   2798= 2799= 2800= 2801= 2802= 2803= 2816= 2817=
U 10A8   2831= 2832= 2839= 2840= 2847= 2848= 2852= 2853=
U 10B0   2855= 2856= 2899= 2900= 2909= 2910= 2912= 2913=
U 10B8   2914= 2915= 2916= 2917= 2928= 2929= 2948= 2949=
U 10C0   2956= 2957= 2961= 2962= 2964= 2965= 3007= 3008=
U 10C8   3023= 3024= 3026= 3027= 3028= 3029= 3035= 3036=
U 10D0   3037= 3038= 3053= 3054= 3074= 3075= 3082= 3083=
U 10D8   3087= 3088= 3090= 3091= 3099= 3100= 3139= 3140=
U 10E0   3148= 3149= 3151= 3152= 3153= 3154= 3155= 3156=
U 10E8   3169= 3170= 3184= 3185= 3192= 3193= 3200= 3201=
U 10F0   3205= 3206= 3208= 3209= 3248= 3249= 3258= 3259=
U 10F8   3261= 3262= 3263= 3264= 3265= 3266= 3277= 3278=
U 1100   1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108   1869: 1880 3297= 3298= 1870: 1871: 1872: 1873:
U 1110   3305= 3306= 3310= 3311= 3313= 3314= 3356= 3357=
U 1118   3372= 3373= 3375= 3376= 3377= 3378= 3384= 3385=
U 1120   3386= 3387= 3402= 3403= 3423= 3424= 3431= 3432=
U 1128   3436= 3437= 3439= 3440= 3448= 3449= 3453= 3454=
U 1130   3458= 1900 1901 1902 1903 1904 1905 1906
U 1138   1916 1917 1920 1921 1922 1923 1924 1925
U 1140   1926 1927 1928 1929 1930 1931 1932 1933
U 1148   1934 1935 1936 1937 1938 1940 1941 1973
U 1150   1974 1980 1981 1982 1993 1965: 1994 1997
U 1158   2002 2005 2015 2016 2017 2025 2026 2027
U 1160   2033 2034 2035 2036 2037 2038 2039 2040
U 1168   2041 2042 2054 2055 2056 2057 2061 2062
U 1170   2073 2074 2075 2076 2079 2091 2092 2093
U 1178   2094 2098 2099 2112 2113 2114 2115 2116
U 1180   2120 2121 2129 2130 2131 2132 2133 2134
U 1188   2135 2136 2137 2138 2139 2140 2141 2142
U 1190   2149 2150 2151 2152 2153 2154 2155 2156
U 1198   2157 2158 2159 2160 2161 2162 2169 2170
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2171 2174 2175 2176 2183 2184 2185 2188
U 11A8  2189 2190 2197 2198 2199 2202 2203 2204
U 11B0  2210 2213 2214 2215 2216 2217 2218 2219
U 11B8  2220 2221 2222 2223 2224 2225 2226 2227
U 11C0  2228 2229 2271 2272 2273 2274 2275 2276
U 11C8  2277 2278 2284 2285 2286 2287 2288 2289
U 11D0  2290 2291 2295 2296 2297 2298 2299 2300
U 11D8  2301 2302 2303 2304 2305 2306 2307 2308
U 11E0  2309 2310 2311 2312 2315 2316 2317 2318
U 11E8  2319 2320 2321 2322 2325 2328 2329 2332
U 11F0  2333 2334 2335 2339 2381 2382 2383 2384
U 11F8  2385 2386 2387 2388 2394 2395 2396 2397
U 1200  2398 2399 2400 2401 2402 2403 2407 2408
U 1208  2409 2410 2411 2412 2413 2414 2415 2416
U 1210  2417 2418 2419 2420 2421 2422 2423 2424
U 1218  2427 2428 2429 2430 2431 2432 2433 2434
U 1220  2437 2440 2441 2444 2445 2446 2447 2451
U 1228  2508 2511 2514 2517 2518 2519 2520 2528
U 1230  2529 2530 2533 2534 2535 2536 2537 2538
U 1238  2539 2540 2541 2542 2547 2548 2549 2550
U 1240  2551 2552 2553 2554 2555 2556 2566 2569
U 1248  2570 2571 2572 2573 2574 2575 2576 2577
U 1250  2578 2583 2584 2585 2586 2587 2588 2589
U 1258  2590 2591 2592 2602 2603 2604 2607 2608
U 1260  2609 2610 2611 2612 2613 2614 2615 2620
U 1268  2621 2622 2623 2624 2625 2626 2627 2628
U 1270  2629 2639 2640 2643 2644 2645 2646 2647
U 1278  2648 2649 2650 2651 2652 2664 2665 2666
U 1280  2667 2668 2669 2670 2671 2672 2673 2685
U 1288  2686 2687 2688 2689 2690 2691 2692 2693
U 1290  2694 2704 2705 2708 2709 2710 2711 2712
U 1298  2713 2714 2715 2716 2717 2727 2728 2729
U 12A0  2732 2733 2734 2735 2736 2737 2738 2739
U 12A8  2740 2741 1966: 2745 2788 2789 2790 2791
U 12B0  2792 2793 2794 2804 2805 2806 2807 2809
U 12B8  2810 2811 2812 2813 2814 2815 2818 2819
U 12C0  2820 2821 2822 2823 2824 2825 2826 2827
U 12C8  2828 2829 2830 2833 2834 2835 2836 2837
U 12D0  2838 2841 2842 2843 2844 2845 2846 2849
U 12D8  2850 2851 2854 2901 2902 2903 2904 2905
U 12E0  2906 2907 2908 2918 2919 2920 2921 2922
U 12E8  2923 2924 2925 2926 2927 2930 2931 2932
U 12F0  2933 2934 2935 2936 2937 2938 2939 2940
U 12F8  2941 2942 2945 2946 2947 2950 2951 2952
U 1300  2953 2954 2955 2958 2959 2960 2963 3009
U 1308  3010 3011 3012 3013 3014 3015 3016 3017
U 1310  3018 3019 3020 3021 3022 3030 3031 3032
U 1318  3033 3034 3039 3040 3041 3042 3043 3044
U 1320  3045 3046 3047 3048 3049 3050 3051 3052
U 1328  3055 3056 3057 3058 3059 3060 3061 3062
U 1330  3063 3064 3065 3066 3067 3068 3071 3072
U 1338  3073 3076 3077 3078 3079 3080 3081 3084
U 1340  3085 3086 3089 3092 3093 3094 3095 3096
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3097 3098 3141 3142 3143 3144 3145 3146
U 1350  3147 3157 3158 3159 3160 3162 3163 3164
U 1358  3165 3166 3167 3168 3171 3172 3173 3174
U 1360  3175 3176 3177 3178 3179 3180 3181 3182
U 1368  3183 3186 3187 3188 3189 3190 3191 3194
U 1370  3195 3196 3197 3198 3199 3202 3203 3204
U 1378  3207 3250 3251 3252 3253 3254 3255 3256
U 1380  3257 3267 3268 3269 3270 3271 3272 3273
U 1388  3274 3275 3276 3279 3280 3281 3282 3283
U 1390  3284 3285 3286 3287 3288 3289 3290 3291
U 1398  3294 3295 3296 3299 3300 3301 3302 3303
U 13A0  3304 3307 3308 3309 3312 3358 3359 3360
U 13A8  3361 3362 3363 3364 3365 3366 3367 3368
U 13B0  3369 3370 3371 3379 3380 3381 3382 3383
U 13B8  3388 3389 3390 3391 3392 3393 3394 3395
U 13C0  3396 3397 3398 3399 3400 3401 3404 3405
U 13C8  3406 3407 3408 3409 3410 3411 3412 3413
U 13D0  3414 3415 3416 3417 3420 3421 3422 3425
U 13D8  3426 3427 3428 3429 3430 3433 3434 3435
U 13E0  3438 3441 3442 3443 3444 3445 3446 3447
U 13E8 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR5E	1016
Used	1016
Remaining	

Total mirrowords used in memory U: 1016
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKARSE.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR5F.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 2544

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2230 TEST 26D AMXC/COND TEST
; 2310 TEST 26E BEN/5 BIT<0> TEST 1
; 2413 TEST 26F BEN/5 BIT<0> TEST 2
; 2926 TEST 270 BEN/5 BIT<1> TEST 1
; 3012 TEST 271 BEN/5 BIT<1> TEST 2
; 3068 TEST 272 FALU A+B TEST 1
; 3174 TEST 273 FALU A+B TEST 2
; 3281 TEST 274 FALU A-B TEST 1
; 3395 TEST 275 FALU A-B TEST 2
; 3512 TEST 276 RESERVED
; 3517 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR5F
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003   ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003   ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003   ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003   ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003   ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003   ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003   ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003   ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003   ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003   ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003   ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003   ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003   ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1046   ;1874 TRPH0: Q_ID[USTACK]
U 1046. 0C00.003C.01E0.0800.0000.1056   ;1875  Q_D,D_Q
U 1056. 0000.003C.8180.3C00.0000.105B   ;1876  ID[USTACK)_D
U 105B. 0C00.003C.01E0.0800.0000.1066   ;1877  Q_D,D_Q
U 1066. 0000.003C.01C0.0A78.0000.106E   ;1878  Q_R[0F]
U 106E. 0001.2024.0180.0800.0010.1109   ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1109. 0000.013C.0180.0800.0000.1002   ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000   ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004, 0000.003D.D9B0.0800.0084.7198  ;1898 =0 SETRCF[.9]
U 1005, 0000.003C.49F0.2C00.0000.1154  ;1899  Q_ID[TBER0]
U 1154, 0001.203C.4DF0.2DB0.0000.1179  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1179, 0001.203C.65F0.2DB8.0000.117A  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 117A, 0001.203C.6DF0.2DD8.0000.117B  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 117B, 0001.203C.75F0.2DE0.0000.117C  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 117C, 0001.203C.79F0.2DC8.0000.117D  ;1904  RC[9]_Q,Q_ID[PARITY]
U 117D, 0001.203C.69F0.2DC0.0000.117E  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 117E, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1192  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 117F, 0000.003C.8580.0800.0084.7180  ;1916 SCOPE: SC_K[.C]
U 1180, 0803.001C.0180.0800.0000.1006  ;1917  ALU_NOT.MASK,D_ALU
U 1006, 0000.003D.7580.3C00.0000.119B  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1007, 0000.003C.65F8.0800.0084.7181  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 1181, 0818.0038.8D80.0800.0000.1182  ;1920  D_K[.1F]    ; ...
U 1182, 0D00.003C.0180.0800.0000.1183  ;1921  D_DAL.SC
U 1183, 0000.003C.3D80.3C00.0000.1184  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 1184, 0818.0038.0580.0800.0000.1185  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 1185, 0D00.003C.0180.0800.0000.1186  ;1924  D_DAL.SC
U 1186, 0F00.003C.3180.3C00.0000.1187  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1187, 0000.003C.0180.0800.0000.1188  ;1926  NOP
U 1188, 0000.003C.3980.3C00.0000.1189  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1189, 0000.003C.2980.3C00.0000.118A  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 118A, 0000.003C.4980.3C00.0000.118B  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 118B, 0818.0038.C180.0800.0000.118C  ;1930  D_K[.FFFF]
U 118C, 0000.003C.6580.3C00.0000.118D  ;1931  ID[SBI.ERR]_D
U 118D, 0F00.003C.6D80.3C00.0000.118E  ;1932  ID[FAULT]_D,D_0
U 118E, 0000.003C.6D80.3C00.0000.118F  ;1933  ID[FAULT]_D
U 118F, 0000.003C.6580.3C00.0000.1190  ;1934  ID[SBI.ERR]_D
U 1190, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 1191, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 1192, 0810.0038.0180.0978.0000.1193  ;1937 ERROR1: D_RC[0F]
U 1193, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 1194, 0810.0038.0180.0978.0000.1195  ;1940 ERROR2: D_RC[0F]
U 1195, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155, 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
           ;1967 ;*
           ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
           ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
           ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
           ;1971 ; TYPING IT.
           ;1972 ;-
U 1196, 0810.0038.4D80.0978.0000.1197  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1197, 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D*K[.FF00],WORD,RETURN1
           ;1975 ;*
           ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
           ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
           ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
           ;1979 ;-
U 1198, 0010.0038.01C0.0978.0000.1199  ;1980 SETRCF: Q_RC[0F]
U 1199, 0019.2034.4DC0.0800.0000.119A  ;1981  Q_Q.AND.K[.FF00]
U 119A, 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
           ;1983 ;
           ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
           ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
           ;1986 ;   28: NAD/28  ; NOP
           ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
           ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
           ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
           ;1990 ;  INITIALIZE ITSELF.
           ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
           ;1992 ;
U 119B, 0818.0038.4580.0800.0000.119C  ;1993 FPINIT: D_K[.8000]
U 119C, 0000.003C.5D80.3C00.0000.1008  ;1994  ID[ACC.CS]_D  ; TURN ON FPA
U 1008, 0818.0039.2D80.0800.0000.11A0  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1009, 0000.003C.5980.3C00.0000.119D  ;1996  ID[ACC.2]_D
U 119D, 0000.00FC.0380.0800.0000.100A  ;1997  TRAP.FPA  ; TRAP FPA TO 28.
U 100A, 0018.0039.1980.0800.0200.11A6  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 100B, 0F00.003C.0180.0800.0000.100C  ;1999  D_0
U 100C, 0000.003D.0180.0800.0000.11A0  ;2000 =0 CALL,J/GENTRA
U 100D, 0000.003C.5980.3C00.0000.119E  ;2001  ID[ACC.2]_D
U 119E, 0000.00FC.0380.0800.0000.1010  ;2002  TRAP.FPA  ; TRAP FPA TO 0
U 1010, 0818.0039.2D80.0800.0000.11A0  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 1011, 0000.003C.5980.3C00.0000.119F  ;2004  ID[ACC.2]_D
U 119F, 0000.00FE.0380.0800.0000.0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
            ;2006 ;
            ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2008 ;%0
            ;2009 ;$ 0000.0000. 0000.0000
            ;2010 ;
            ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2014 ;
U 11A0, 0000.003C.65F8.0800.0084.71A1  ;2015 GENTRA: SC_K[.10],Q_0
U 11A1, 0D00.003C.8D80.0800.0084.71A2  ;2016  D_DAL.SC,SC_K[.1F]
U 11A2, 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2018 ;
            ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2021 ;
            ;2022 =0
U 1012, 0F00.003D.0180.0800.0000.11A0  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 1013, 0000.003C.5980.3C00.0000.11A3  ;2024  ID[ACC.2]_D
U 11A3, 0000.00FC.0380.0800.0000.11A4  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 11A4, 0000.003C.0180.0800.0000.11A5  ;2026  NOP
U 11A5, 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
            ;2028 ;
            ;2029 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2030 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2031 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2032 ;
U 11A6, 2000.003C.0180.0800.0000.11A7  ;2033 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 11A7, 3000.003C.0180.6000.0000.11A8  ;2034  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 11A8, 0000.003C.0180.F800.0000.11A9  ;2035 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 11A9, 0000.003C.0180.F800.0000.11AA  ;2036  MCT/ALLOW.IB.READ
U 11AA, 0000.003C.0180.F800.0000.11AB  ;2037  MCT/ALLOW.IB.READ
U 11AB, 0000.003C.0180.F800.0000.11AC  ;2038  MCT/ALLOW.IB.READ
U 11AC, 0000.003C.0180.F800.0000.11AD  ;2039  MCT/ALLOW.IB.READ
U 11AD, 1000.003C.0180.F800.0000.11AE  ;2040  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 11AE, 0000.003E.0180.0800.0000.0001  ;2041  RETURN1
U 11AF, 3000.003C.0180.0800.0000.11A8  ;2042 WAITIB: IBC/START,J/IBW
            ;2043 ;
            ;2044 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
            ;2045 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
            ;2046 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
            ;2047 ; RC[8]  A OPERAND BITS <63:32>
            ;2048 ; RC[7]  A OPERAND BITS <31:00>
            ;2049 ; RC[6]  B OPERAND BITS <63:32>
            ;2050 ; RC[5]  B OPERAND BITS <31:00>
            ;2051 ; RC[0A]  RESULT BITS <63:32>
            ;2052 ; RC[9]  RESULT BITS <31:00>
            ;2053 ;
U 11B0, 0810.0038.0180.0938.0000.11B1  ;2054 ADDSIM: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 11B1, 0010.0038.01C0.0928.0000.11B2  ;2055  Q_RC[5]
U 11B2, 001D.0014.0180.09C8.0010.11B3  ;2056  ALU_D+Q,RC[9]_ALU,CLK.UBCC ; AND SAVE THE CARRY
U 11B3, 0810.1B38.0180.0940.0000.100E  ;2057  D_RC[8],ALU.C?
U 100E, 0000.003C.0180.0800.0000.11B4  ;2058 =1110 J/ADSM1   ; NO CARRY
U 100F, 0819.0014.0580.0800.0000.11B4  ;2059  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
            ;2060     ; RESULT
```

```
U 11B4, 0010.0038.01C0.0930.0000.11B5  ;2061 ADSM1: Q_RC[6]
U 11B5, 001D.0016.0180.09D0.0000.0001  ;2062  ALU_D+Q,RC[0A]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
          ;2063      ; AND RETURN.
          ;2064 ;
          ;2065 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU SUBTRACT. IT PERFORMS A
          ;2066 ; DOUBLE PRECISION SUB ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
          ;2067 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
          ;2068 ; RC[8]  A OPERAND BITS <63:32>
          ;2069 ; RC[7]  A OPERAND BITS <31:00>
          ;2070 ; RC[6]  B OPERAND BITS <63:32>
          ;2071 ; RC[5]  B OPERAND BITS <31:00>
          ;2072 ;
U 11B6, 0010.0038.01C0.0938.0000.11B7  ;2073 SUBSIM: Q_RC[7]   ; DO LO SUB
U 11B7, 0810.0038.0180.0928.0000.11B8  ;2074  D_RC[5]
U 11B8, 001D.0000.0180.09C8.0010.11B9  ;2075  ALU_D-Q,RC[9]_ALU,CLK.UBCC ; SAVE BORROW
U 11B9, 0010.1B38.01C0.0940.0000.101E  ;2076  Q_RC[8],ALU.C?
U 101E, 0019.2014.05C0.0800.0000.101F  ;2077 =1110 ALU_Q+K[.1],Q_ALU ; SUB BORROW
U 101F, 0810.0038.0180.0930.0000.11BA  ;2078 SBSM1: D_RC[6]
U 11BA, 001D.0002.0180.09D0.0000.0001  ;2079  ALU_D-Q,RC[0A]_ALU,RETURN1 ; DO UPPER HALF OF SUB.
          ;2080 ;
          ;2081 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
          ;2082 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], R[0D]
          ;2083 ; AND R[0C] AND LEAVES THE RESULT IN RC[8] AND RC[7].
          ;2084 ; RC[8]  A OPERAND BITS <63:32>
          ;2085 ; RC[7]  A OPERAND BITS <31:00>
          ;2086 ; R[0C]  B OPERAND BITS <63:32>
          ;2087 ; R[0D]  B OPERAND BITS <31:00>
          ;2088 ; RC[8]  RESULT BITS <63:32>
          ;2089 ; RC[7]  RESULT BITS <31:00>
          ;2090 ;
U 11BB, 0810.0038.0180.0938.0000.11BC  ;2091 INCA: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 11BC, 0000.003C.01C0.0A68.0000.11BD  ;2092  Q_R[0D]
U 11BD, 001D.0014.0180.09B8.0010.11BE  ;2093  ALU_D+Q,RC[7]_ALU,CLK.UBCC ; AND SAVE THE CARRY
U 11BE, 0810.1B38.0180.0940.0000.102E  ;2094  D_RC[8],ALU.C?
U 102E, 0000.003C.0180.0800.0000.11BF  ;2095 =1110 J/INCA1   ; NO CARRY
U 102F, 0819.0014.0580.0800.0000.11BF  ;2096  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
          ;2097      ; RESULT
U 11BF, 0000.003C.01C0.0A60.0000.11C0  ;2098 INCA1: Q_R[0C]
U 11C0, 001D.0016.0180.09C0.0000.0001  ;2099  ALU_D+Q,RC[8]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
          ;2100      ; AND RETURN.
          ;2101 ;
          ;2102 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
          ;2103 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[5], RC[6], R[0D]
          ;2104 ; AND R[0C] AND LEAVES THE RESULT IN RC[6] AND RC[5].
          ;2105 ; RC[6]  A OPERAND BITS <63:32>
          ;2106 ; RC[5]  A OPERAND BITS <31:00>
          ;2107 ; R[0C]  B OPERAND BITS <63:32>
          ;2108 ; R[0D]  B OPERAND BITS <31:00>
          ;2109 ; RC[6]  RESULT BITS <63:32>
          ;2110 ; RC[5]  RESULT BITS <31:00>
          ;2111 ;
U 11C1, 0810.0038.0180.0928.0000.11C2  ;2112 INCB: D_RC[5]   ; FIRST ADD LO FRACTIONS
U 11C2, 0000.003C.01C0.0A68.0000.11C3  ;2113  Q_R[0D]
U 11C3, 0019.2024.59C0.0800.0000.11C4  ;2114  ALU_Q[ANDNOT]K[.7F],Q_ALU
U 11C4, 001D.0014.0180.09A8.0010.11C5  ;2115  ALU_D+Q,RC[5]_ALU,CLK.UBCC ; AND SAVE THE CARRY
```

```
U 11C5, 0810.1B38.0180.0930.0000.103E  ;2116  D_RC[6],ALU.C?
U 103E, 0000.003C.0180.0800.0000.11C6  ;2117 =1110 J/INCB1   ; NO CARRY
U 103F, 0819.0014.0580.0800.0000.11C6  ;2118  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
          ;2119      ; RESULT
U 11C6, 0000.003C.01C0.0A60.0000.11C7  ;2120 INCB1: Q_R[0C]
U 11C7, 001D.0016.0180.09B0.0000.0001  ;2121  ALU_D+Q,RC[6]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
          ;2122      ; AND RETURN.
          ;2123 ;
          ;2124 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2125 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2126 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2127 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2128 ;
U 11C8, 0000.003C.01C0.0A28.0000.11C9  ;2129 UNSEQA: Q_R[5]   ; GET FRACTION INTO D AND Q.
U 11C9, 0800.003C.B980.0A20.0084.71CA  ;2130  D_R[4],SC_K[.19]
U 11CA, 0D00.003C.0180.0800.0000.11CB  ;2131  D_DAL.SC
U 11CB, 0000.003C.65E0.0800.0084.71CC  ;2132  Q_D,SC_K[.10]
U 11CC, 0D00.003C.0180.0800.0000.11CD  ;2133  D_DAL.SC
U 11CD, 0001.003C.0180.0AB8.0000.11CE  ;2134  R[7]_D   ; SAVE LEAST SIGNIFICANT RESULT, LO
U 11CE, 0800.003C.01C0.0A20.0000.11CF  ;2135  D_R[4],Q_R[4]
U 11CF, 0819.0024.5980.0800.0000.11D0  ;2136  ALU_D[ANDNOT]K[.7F],D_ALU
U 11D0, 0100.003C.5D88.0800.0084.71D1  ;2137  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 11D1, 0D00.003C.0180.0800.0000.11D2  ;2138  D_DAL.SC
U 11D2, 0001.003C.0180.0AB0.0084.71D3  ;2139  R[6]_D,SC_K[.8]
U 11D3, 0818.0034.59F8.0A28.0000.11D4  ;2140  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 11D4, 0D00.003C.01C0.0A30.0000.11D5  ;2141  D_DAL.SC,Q_R[6]
U 11D5, 001D.0032.0180.0AB0.0000.0001  ;2142  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2143 ;
          ;2144 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2145 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2146 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2147 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2148 ;
U 11D6, 0000.003C.01C0.0A28.0000.11D7  ;2149 UNSEQB: Q_R[5]   ; GET FRACTION INTO D AND Q.
U 11D7, 0800.003C.B980.0A20.0084.71D8  ;2150  D_R[4],SC_K[.19]
U 11D8, 0D00.003C.0180.0800.0000.11D9  ;2151  D_DAL.SC
U 11D9, 0000.003C.65E0.0800.0084.71DA  ;2152  Q_D,SC_K[.10]
U 11DA, 0D00.003C.0180.0800.0000.11DB  ;2153  D_DAL.SC
U 11DB, 0001.003C.0180.0AB8.0000.11DC  ;2154  R[7]_D   ; SAVE LEAST SIGNIFICANT RESULT, LO
U 11DC, 0800.003C.01C0.0A20.0000.11DD  ;2155  D_R[4],Q_R[4]
U 11DD, 0819.0024.5980.0800.0000.11DE  ;2156  ALU_D[ANDNOT]K[.7F],D_ALU
U 11DE, 0100.003C.5D88.0800.0084.71DF  ;2157  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 11DF, 0D00.003C.0180.0800.0000.11E0  ;2158  D_DAL.SC
U 11E0, 0001.003C.0180.0AB0.0084.71E1  ;2159  R[6]_D,SC_K[.8]
U 11E1, 0818.0034.59F8.0A28.0000.11E2  ;2160  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 11E2, 0D00.003C.01C0.0A30.0000.11E3  ;2161  D_DAL.SC,Q_R[6]
U 11E3, 001D.0032.0180.0AB0.0000.0001  ;2162  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2163 ;
          ;2164 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          ;2165 ; IN RC[0A] AND BITS <31:00> IN RC[9]) INTO THE FLOATING POINT FORMAT
          ;2166 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN RC[0E]
          ;2167 ; AND BITS <31:00> IN RC[0D].
          ;2168 ;
U 11E4, 0810.0038.0180.0950.0000.11E5  ;2169 UNSRES: D_RC[0A]
U 11E5, 0010.0038.01C0.0948.0000.11E6  ;2170  Q_RC[9]
```

```
U 11E6, 0001,003C,0180,0AA0,0000,1014  ;2171  R[4]_D
U 1014, 0001,203D,0180,0AA8,0000,11C8  ;2172 =0 R[5]_Q,CALL,J/UNSEQA
U 1015, 0800,003C,0180,0A30,0000,11E7  ;2173  D_R[6]
U 11E7, 0000,003C,01C0,0A38,0000,11E8  ;2174  Q_R[7]
U 11E8, 0001,003C,0180,09F0,0000,11E9  ;2175  RC[0E]_D
U 11E9, 0001,203E,0180,09E8,0000,0001  ;2176  RC[0D]_Q,RETURN1
          ;2177 ;
          ;2178 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          ;2179 ; IN RC[8] AND BITS <31:00> IN RC[7]) INTO THE FLOATING POINT FORMAT
          ;2180 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[8]
          ;2181 ; AND BITS <31:00> IN R[9].
          ;2182 ;
U 11EA, 0810,0038,0180,0940,0000,11EB  ;2183 UNSAOP: D_RC[8]
U 11EB, 0010,0038,01C0,0938,0000,11EC  ;2184  Q_RC[7]
U 11EC, 0001,003C,0180,0AA0,0000,1016  ;2185  R[4]_D
U 1016, 0001,203D,0180,0AA8,0000,11C8  ;2186 =0 R[5]_Q,CALL,J/UNSEQA
U 1017, 0800,003C,0180,0A30,0000,11ED  ;2187  D_R[6]
U 11ED, 0000,003C,01C0,0A38,0000,11EE  ;2188  Q_R[7]
U 11EE, 0001,003C,0180,0AC0,0000,11EF  ;2189  R[8]_D
U 11EF, 0001,203E,0180,0AC8,0000,0001  ;2190  R[9]_Q,RETURN1
          ;2191 ;
          ;2192 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          ;2193 ; IN RC[6] AND BITS <31:00> IN RC[5]) INTO THE FLOATING POINT FORMAT
          ;2194 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[0A]
          ;2195 ; AND BITS <31:00> IN R[0B].
          ;2196 ;
U 11F0, 0810,0038,0180,0930,0000,11F1  ;2197 UNSBOP: D_RC[6]
U 11F1, 0010,0038,01C0,0928,0000,11F2  ;2198  Q_RC[5]
U 11F2, 0001,003C,0180,0AA0,0000,1018  ;2199  R[4]_D
U 1018, 0001,203D,0180,0AA8,0000,11C8  ;2200 =0 R[5]_Q,CALL,J/UNSEQA
U 1019, 0800,003C,0180,0A30,0000,11F3  ;2201  D_R[6]
U 11F3, 0000,003C,01C0,0A38,0000,11F4  ;2202  Q_R[7]
U 11F4, 0001,003C,0180,0AD0,0000,11F5  ;2203  R[0A]_D
U 11F5, 0001,203E,0180,0AD8,0000,0001  ;2204  R[0B]_Q,RETURN1
          ;2205 ;
          ;2206 ; THIS ROUTINE IS CALLED BY THE FALU TESTS TO LOAD LA, SA,
          ;2207 ; LB, SB, AR AND BR. FALU/A.B FUNCTION IS EXECUTED
          ;2208 ; AND THE RESULTS RETRIEVED.
          ;2209 ;
U 11F6, 0818,0038,3180,0800,0000,101A  ;2210 EXEC: D_K[.40]
U 101A, 0819,0031,6180,0800,0000,11A0  ;2211 =0 ALU_D[OR]K[.F],D_ALU,CALL,J/GENTRA
U 101B, 0000,003C,5980,3C00,0000,11F7  ;2212  ID[ACC.2]_D
U 11F7, 0000,00FC,0380,0800,0000,11F8  ;2213  TRAP.FPA  ; TRAP TO FPA 4F
U 11F8, 0000,003C,0180,0A10,0000,11F9  ;2214  LAB_R[2]  ; AT FPA 4F
U 11F9, 0000,007C,0180,0A10,0000,11FA  ;2215  LAB_R[2],CPSYNC  ; AT FPA 4F
U 11FA, 0000,003C,0180,0A18,0000,11FB  ;2216  LAB_R[3]  ; AT FPA 39
U 11FB, 0000,007C,0180,0A18,0000,11FC  ;2217  LAB_R[3],CPSYNC  ; AT FPA 39
U 11FC, 0000,003C,0180,0A40,0000,11FD  ;2218  LAB_R[8]  ; AT FPA 3A
U 11FD, 0000,007C,0180,0A40,0000,11FE  ;2219  LAB_R[8],CPSYNC  ; AT FPA 3A
U 11FE, 0000,003C,0180,0A48,0000,11FF  ;2220  LAB_R[9]  ; AT FPA 3B
U 11FF, 0000,007C,0180,0A48,0000,1200  ;2221  LAB_R[9],CPSYNC  ; AT FPA 3B
U 1200, 0000,003C,0180,0A50,0000,1201  ;2222  LAB_R[0A]  ; AT FPA 3C
U 1201, 0000,007C,0180,0A50,0000,1202  ;2223  LAB_R[0A],CPSYNC ; AT FPA 3C
U 1202, 0000,003C,0180,0A58,0000,1203  ;2224  LAB_R[0B]  ; AT FPA 3D
U 1203, 0000,007C,0180,0A58,0000,1204  ;2225  LAB_R[0B],CPSYNC ; AT FPA 3D
```

```
U 1204, 0A00,003C,0180,0800,0000,1205  ;2226  D_ACC    ; AT FPA 3E, GET RESULT HI
U 1205, 0000,003C,01D8,0800,0000,1206  ;2227  Q_ACC    ; AT FPA 3F, GET RESULT LO
U 1206, 0001,003C,0180,09E0,0000,1207  ;2228  RC[0C]_D
U 1207, 0001,203E,0180,09D8,0000,0001  ;2229  RC[0B]_Q,RETURN1
```

```
          ;2230  .PAGE "TEST 26D AMXC/COND TEST"
          ;2231 ;*****************************************************************
          ;2232 ;**
          ;2233 ; AMXC/COND TEST
          ;2234 ;
          ;2235 ; TEST DESCRIPTION
          ;2236 ; THIS IS A TEST OF THE AMXC CONTROL FUNCTION WHICH GATES EITHER
          ;2237 ; LA OR LB (WHICH EVER IS GREATER) THRU THE AMX.
          ;2238 ; EVERY COMBINATION OF LA VERSUS LB VALUES IS TRIED.
          ;2239 ;
          ;2240 ; LOGIC DESCRIPTION
          ;2241 ;
          ;2242 ; FPA UCODE USED
          ;2243 ; LOCATION CONTENTS
          ;2244 ; 9C:  BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/9D
          ;2245 ; 9D:  BSC/R,SCR/CPU,EAC/LDB,WAIT,NAD/9E
          ;2246 ; 9E:  AMXC/COND,EALUC/A,EAC/LDPR,NAD/9F
          ;2247 ; 9F:  AMXC/PR,EALUC/A,BSC/NH,NAD/PSTALL
          ;2248 ;
          ;2249 ; ERROR DESCRIPTION
          ;2250 ; EXPECTED RESULT <14:07>
          ;2251 ; GOT RESULT <14:07>
          ;2252 ; CONTENTS OF LA <14:07>
          ;2253 ; CONTENTS OF LB <14:07>
          ;2254 ;
          ;2255 ; SYNC POINT DESCRIPTION
          ;2256 ;
          ;2257 ;--
          ;2258 ;*****************************************************************
          ;2259 =0
U 101C, 0000.003D,0180,0800,0000,117F  ;2260 AMXC: NEWTST
U 101D, 0018.0038,1180,09F8,0000,1208  ;2261  RC[0F]_K[.4]
U 1208, 0818.0038,4980,0800,0000,1209  ;2262  D_K[.FF]
U 1209, 0000.003C,5DF8,0800,0084,720A  ;2263  SC_K[.7],Q_0
U 120A, 0D00.003C,0180,0800,0000,120B  ;2264  D_DAL.SC
U 120B, 0001.003C,0180,0AA8,0000,120C  ;2265  R[5]_D    ; GENERATE A MASK, 00007F800
U 120C, 0018.0038,4980,0A88,0000,120D  ;2266  R[1]_K[.FF]  ; INITIALIZE LA
U 120D, 0018.0038,4980,0A90,0000,1020  ;2267  R[2]_K[.FF]  ; INITIALIZE LB
U 1020, 0000.003D,0180,0800,0000,1191  ;2268 =0 ERLOOP
U 1021, 0000.003C,0180,0800,0000,1022  ;2269  NOP
          ;2270 =0
U 1022, 0000.003D,0180,0800,0000,119B  ;2271 AMXC1: CALL,J/FPINIT
U 1023, 0818.0038,4180,0800,0000,120E  ;2272  D_K[.80]  ; GENERATE THE ADDRESS OF THE
U 120E, 0819.0030,7D80,0800,0000,1024  ;2273  ALU_D[OR]K[.18],D_ALU ; FPA UCODE.
U 1024, 0819.0015,1180,0800,0000,11A0  ;2274 =0 ALU_D+K[.4],D_ALU,CALL,J/GENTRA
U 1025, 0000.003C,5980,3C00,0000,120F  ;2275  ID[ACC.2]_D
U 120F, 0800.003C,5DF8,0A08,0084,7210  ;2276  D_R[1],Q_0,SC_K[.7] ; CALCULATE THE LA AND LB VALUES
U 1210, 0D00.003C,0180,0800,0000,1211  ;2277  D_DAL.SC
U 1211, 0001.003C,0180,0A98,0000,1212  ;2278  R[3]_D
U 1212, 0001.003C,0180,09E0,0000,1213  ;2279  RC[0C]_D
U 1213, 0800.003C,0180,0A08,0000,1214  ;2280  D_R[1]
U 1214, 0D00.003C,0180,0800,0000,1215  ;2281  D_DAL.SC
U 1215, 0001.003C,0180,09D8,0000,1216  ;2282  RC[0B]_D
U 1216, 0001.003C,0180,0AA0,0000,1217  ;2283  R[4]_D
U 1217, 0800.003C,0180,0A18,0000,1218  ;2284  D_R[3]
```

```
U 1218. 0000.003C.01C0.0A20.0000.1219  ;2285  Q_R[4]
U 1219. 001D.0000.0180.0800.0010.121A  ;2286  ALU_D-Q,CLK.UBCC ; SEE WHICH IS GREATER LA OR LB?
U 121A. 0001.3B3C.0180.09F0.0000.1047  ;2287  ALU.N?,RC[0E]_Q
U 1047. 0001.003C.0180.09F0.0000.104F  ;2288 =0111 RC[0E]_D
U 104F. 0000.00FC.0380.0800.0000.121B  ;2289  TRAP.FPA  ; TRAP TO FPA 9C
U 121B. 0000.003C.0180.0A18.0000.121C  ;2290  LAB_R[3]  ; AT FPA 9C
U 121C. 0000.007C.0180.0A18.0000.121D  ;2291  LAB_R[3],CPSYNC  ; AT FPA 9C
U 121D. 0000.003C.0180.0A20.0000.121E  ;2292  LAB_R[4]  ; AT FPA 9D
U 121E. 0000.007C.0180.0A20.0000.121F  ;2293  LAB_R[4],CPSYNC  ; AT FPA 9D
U 121F. 0000.003C.0180.0800.0000.1220  ;2294  NOP  ; AT FPA 9E
U 1220. 0A00.003C.0180.0800.0000.1221  ;2295  D_ACC  ; AT FPA 9F, GET RESULT
U 1221. 080D.0034.0180.0A28.0000.1222  ;2296  ALU_D[AND]R[5],D_ALU ; MASK EXPONENT.
U 1222. 0001.003C.0180.09E8.0000.1223  ;2297  RC[0D]_D
U 1223. 0010.0038.01C0.0970.0000.1224  ;2298  Q_RC[0E]
U 1224. 001D.0000.0180.0800.0010.1225  ;2299  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 1225. 0000.013C.0180.0800.0000.1026  ;2300  Z?
U 1026. 0000.003D.0180.0800.0000.1192  ;2301 =0 ERROR1  ; RESULT INCORRECT.
U 1027. 0000.003C.0180.0A10.0010.1226  ;2302  ALU_R[2],CLK.UBCC
U 1226. 0000.013C.0180.0800.0000.1028  ;2303  Z?
U 1028. 0018.0000.0580.0A90.0000.1022  ;2304 =0 R[2]_LA-K[.1],J/AMXC1 ; DECRIMENT LB AND CONTINUE.
U 1029. 0018.0038.4980.0A90.0000.1227  ;2305  R[2]_K[.FF]
U 1227. 0000.003C.0180.0A08.0010.1228  ;2306  ALU_R[1],CLK.UBCC
U 1228. 0000.013C.0180.0800.0000.102A  ;2307  Z?
U 102A. 0018.0000.0580.0A88.0000.1022  ;2308 =0 R[1]_LA-K[.1],J/AMXC1 ; DECREMENT LA AND CONTINUE
U 102B. 0000.003C.0180.0800.0000.102C  ;2309  NOP
```

```
                :2310  .PAGE "TEST 26E BEN/5 BIT<0> TEST 1"
                :2311  ;*****************************************************************
                :2312  ;**
                :2313  ; BEN/5 BIT<0> TEST 1
                :2314  ;
                :2315  ; TEST DESCRIPTION
                :2316  ; THIS IS A TEST OF BEN/5 BIT<0> WHICH IS EVUIVALENT
                :2317  ; TO COMPLEMENT AR OR COMPLEMENT BR. ALL THE CONDITIONS
                :2318  ; FOR A COMPLEMENT ARE ASSERTED WITH EVERY POSIBLE
                :2319  ; COMBINATION OF VALUES FOR LA AND LB. EACH LA AND LB
                :2320  ; COMBINATION IS TRIED AND THE BRANCH IS TESTED TO INSURE
                :2321  ; THAT IT IS ASSERTED OR DEASSERTED APPROPRIATELY.
                :2322  ;
                :2323  ; LOGIC DESCRIPTION
                :2324  ;
                :2325  ; FPA UCODE USED
                :2326  ; LOCATION CONTENTS
                :2327  ; 36:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/37
                :2328  ; 37:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/8C
                :2329  ; 8C:  FADC/0,NAD/38
                :2330  ; 38:  FADC/A.B,BEN/5,NAD/6
                :2331  ;
                :2332  ; ERROR DESCRIPTION
                :2333  ; EXPECTED NAD
                :2334  ; GOT NAD
                :2335  ; CONTENTS OF LA <07:00>
                :2336  ; CONTENTS OF LB <07:00>
                :2337  ; EXPONENT DIFFERENCE, ABSOLUTE VALUE OF LA-LB, <07:00>
                :2338  ;
                :2339  ; SYNC POINT DESCRIPTION
                :2340  ;
                :2341  ;--
                :2342  ;*****************************************************************
                :2343  =0
U 102C. 0000.003D.0180.0800.0000.117F   :2344 B5T1: NEWTST
U 102D. 0818.0038.1180.0800.0000.1229   :2345  D_K[.4]
U 1229. 0019.0014.0580.09F8.0000.122A   :2346  ALU_D+K[.1],RC[0F]_ALU
U 122A. 0018.0038.4980.0A88.0000.122B   :2347  R[1]_K[.FF]  ; INITIALIZE LA VALUE
U 122B. 0018.0038.4980.0A90.0000.1030   :2348  R[2]_K[.FF]  ; INITIALIZE LB VALUE
U 1030. 0000.003D.0180.0800.0000.1191   :2349 =0 ERLOOP
U 1031. 0000.003C.0180.0800.0000.1032   :2350  NOP
                :2351  =0
U 1032. 0000.003D.0180.0800.0000.119B   :2352 B5T1A: CALL,J/FPINIT
U 1033. 0018.0038.2180.0800.0200.1034   :2353  VA_K[.14]
U 1034. 0000.003D.0180.0800.0000.11A6   :2354 =0 CALL,J/LOADIB  ; LOAD ADDD INTO IB.
U 1035. 0818.0038.7980.0800.0000.1036   :2355  D_K[.30]  ; GENERATE THE ADDRESS OF
U 1036. 0819.0031.D580.0800.0000.11A0   :2356 =0 ALU_D[OR]K[.6],D_ALU,CALL,J/GENTRA ; THE FPA UCODE
U 1037. 0000.003C.5980.3C00.0000.122C   :2357  ID[ACC.2]_D
U 122C. 0818.0038.4180.0800.0000.1038   :2358  D_K[.80]  ; GENERATE ADDRESS 8C SO THAT
U 1038. 0819.0031.8580.0800.0000.11A0   :2359 =0 ALU_D[OR]K[.C],D_ALU,CALL,J/GENTRA ; THIS TEST CAN TRAP THERE.
U 1039. 0001.003C.0180.0AD0.0000.122D   :2360  R[0A]_D
U 122D. 0800.003C.5DF8.0A08.0084.722E   :2361  D_R[1],Q_0,SC_K[.7] ; GENERATE THE LA AND LB VALUES.
U 122E. 0001.003C.0180.09E0.0000.122F   :2362  RC[0C]_D
U 122F. 0D00.003C.0180.0800.0000.1230   :2363  D_DAL,SC
U 1230. 0019.0030.4580.0A98.0000.1231   :2364  ALU_D[OR]K[.8000],R[3]_ALU
```

```
U 1231. 0800.003C.0180.0A10.0000.1232  ;2365  D_R[2]
U 1232. 0001.003C.0180.09D8.0000.1233  ;2366  RC[0B]_D
U 1233. 0D00.003C.0180.0800.0000.1234  ;2367  D_DAL.SC
U 1234. 0001.003C.0180.0AA0.0000.1235  ;2368  R[4]_D
U 1235. 0800.003C.0180.0A08.0000.1236  ;2369  D_R[1]   ; CALCULATE THE EXPECTED RESULT.
U 1236. 0000.003C.01C0.0A10.0000.1237  ;2370  Q_R[2]
U 1237. 081D.0000.0180.0800.0010.1238  ;2371  ALU_D-Q,D_ALU,CLK.UBCC
U 1238. 0000.1B3C.01F8.0800.0000.1057  ;2372  ALU.N?,Q_0
U 1057. 0001.003C.0180.09D0.0000.1239  ;2373 =0111 RC[0A]_D,J/B5T1B
U 105F. 081D.2000.0180.09D0.0000.1239  ;2374  ALU_Q-D,RC[0A]_ALU,D_ALU
U 1239. 0018.0038.79C0.0800.0000.123A  ;2375 B5T1B: Q_K[.30]
U 123A. 0019.2014.01C0.0800.0000.123B  ;2376  ALU_Q+K[.8],Q_ALU
U 123B. 001D.0034.0180.0800.0010.123C  ;2377  ALU_D[AND]Q,CLK.UBCC
U 123C. 0000.013C.0180.0800.0000.103A  ;2378  Z?
U 103A. 0018.0038.5D80.09F0.0000.123D  ;2379 =0 RC[0E]_K[.7],J/B5T1C
U 103B. 0018.0038.D580.09F0.0000.123D  ;2380  RC[0E]_K[.6]
U 123D. 0000.00FC.0380.0800.0000.123E  ;2381 B5T1C: TRAP.FPA  ; TRAP TO FPA 36
U 123E. 0800.003C.0180.0A50.0000.123F  ;2382  D_R[0A]   ; AT FPA 36, SET UP FOR TRAP TO 8C
U 123F. 0000.003C.5980.3C00.0000.1240  ;2383  ID[ACC.2]_D  ; AT FPA 36
U 1240. 0000.003C.0180.0A18.0000.1241  ;2384  LAB_R[3]  ; AT FPA 36
U 1241. 0000.007C.0180.0A18.0000.1242  ;2385  LAB_R[3],CPSYNC  ; AT FPA 36
U 1242. 0000.003C.0180.0A20.0000.1243  ;2386  LAB_R[4]  ; AT FPA 37
U 1243. 0000.007C.0180.0A20.0000.1244  ;2387  LAB_R[4],CPSYNC  ; AT FPA 37
U 1244. 0000.00FC.0380.0800.0000.1245  ;2388  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 1245. 0000.003C.0180.0800.0000.1246  ;2389  NOP   ; AT FPA 8C
U 1246. 0000.003C.59F0.2C00.0000.1247  ;2390  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 1247. 0019.2034.81C0.09E8.0000.1248  ;2391  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1248. 0810.0038.0180.0970.0000.1249  ;2392  D_RC[0E]
U 1249. 001D.0000.0180.0800.0010.124A  ;2393  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT
U 124A. 0000.013C.0180.0800.0000.103C  ;2394  Z?
U 103C. 0000.003D.0180.0800.0000.1192  ;2395 =0 ERROR1   ; BEN 5 FAILED
U 103D. 0000.003C.0180.0A10.0010.124B  ;2396  ALU_R[2],CLK.UBCC
U 124B. 0000.013C.0180.0800.0000.1040  ;2397  Z?
U 1040. 0018.0000.0580.0A90.0000.1032  ;2398 =0 R[2]_LA-K[.1],J/B5T1A ; DECREMENT LB AND CONTINUE
U 1041. 0018.0038.4980.0A90.0000.124C  ;2399  R[2]_K[.FF]
U 124C. 0000.003C.0180.0A08.0010.124D  ;2400  ALU_R[1],CLK.UBCC
U 124D. 0000.013C.0180.0800.0000.1042  ;2401  Z?
U 1042. 0018.0000.0580.0A88.0000.1032  ;2402 =0 R[1]_LA-K[.1],J/B5T1A ; DECREMENT LA AND CONTINUE.
U 1043. 0000.003C.0180.0800.0000.1044  ;2403  NOP
          ;2404 ;
          ;2405 ;%14
          ;2406 ;$ 5060,0050
          ;2407 ;%20
          ;2408 ;$ 5060,0050
          ;2409 ;$ 5040,0050
          ;2410 ;$ 5075,0050
          ;2411 ;$ 5055,0050
          ;2412 ;
```

```
;2413  .PAGE "TEST 26F BEN/5 BIT<0> TEST 2"
;2414 ;*****************************************************************
;2415 ;**
;2416 ; BEN/5 BIT<0> TEST 2
;2417 ;
;2418 ; TEST DESCRIPTION
;2419 ; THIS IS A TEST OF BEN/5 BIT<0> WHICH IS ESSENTIALLY THE
;2420 ; OR OF THE COMPLEMENT AR AND THE COMPLEMENT BR SIGNALS.
;2421 ; THE CONDITIONS TESTED HERE ARE LISTED BELOW:
;2422 ;
;2423 ;  SUBTEST LA SA LB SB INST FADC<3> NAD
;2424 ; 1 39 0 1 0 ADDD 0 6
;2425 ; 2 1 0 39 0 ADDD 0 6
;2426 ; 3 39 1 1 0 ADDD 0 7
;2427 ; 4 1 0 39 1 ADDD 0 7
;2428 ; 5 39 1 1 0 ADDF 0 6
;2429 ; 6 1 0 39 1 ADDF 0 6
;2430 ; 7 39 1 1 0 ADDD 1 6
;2431 ; 8 1 0 39 1 ADDD 1 6
;2432 ; 9 39 1 39 0 POLYD 0 7
;2433 ; A 39 0 39 0 POLYD 0 6
;2434 ; B 39 1 39 0 POLYF 0 6
;2435 ; C 39 0 39 1 POLYD 1 6
;2436 ;
;2437 ; IN THIS TABLE
;2438 ;  INST  IS THE OPCODE IN THE IB
;2439 ;  FADC<3>  IS BIT 3 OF THE FPA UCODE INSTRUCTION FIELD FADC
;2440 ;    IN THE TWO STATES PREVIOUS TO ISSUING
;2441 ;    THE BEN/5.
;2442 ;  NAD  IS THE EXPECTED NAD GENERATED BY THE CONDITIONS
;2443 ;    AND BEN/5,NAD/6.
;2444 ;
;2445 ; LOGIC DESCRIPTION
;2446 ;
;2447 ; FPA UCODE USED
;2448 ; LOCATION CONTENTS
;2449 ; 36:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/37
;2450 ; 37:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/8C
;2451 ; 8C:  FADC/0,NAD/38
;2452 ; 38:  FADC/A,B,BEN/5,NAD/6
;2453 ; 60:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/61
;2454 ; 61:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/8D
;2455 ; 8D:  FADC/8,NAD/62
;2456 ; 62:  FADC/0,BEN/5,NAD/6
;2457 ;
;2458 ; ERROR DESCRIPTION
;2459 ; EXPECTED NAD
;2460 ; GOT NAD
;2461 ;
;2462 ; SYNC POINT DESCRIPTION
;2463 ;
;2464 ;--
;2465 ;*****************************************************************
;2466 =0
U 1044, 0000,003D,0180,0800,0000,117F  ;2467 B5T2: NEWTST
```

```
U 1045. 0018.0038.0980.09F8.0000.124E  ;2468  RC[0F]_K[.2]
U 124E. 0818.0038.7980.0800.0000.124F  ;2469  D_K[.30]  ; GENERATE SOME CONSTANTS.
U 124F. 0819.0030.D980.0800.0000.1250  ;2470  ALU_D[OR]K[.9],D_ALU
U 1250. 0000.003C.5DF8.0800.0084.7251  ;2471  SC_K[.7],Q_0
U 1251. 0D00.003C.0180.0800.0000.1252  ;2472  D_DAL.SC
U 1252. 0001.003C.0180.0A98.0000.1253  ;2473  R[3]_D
U 1253. 0019.0030.4580.0AA0.0000.1254  ;2474  ALU_D[OR]K[.8000],R[4]_ALU
U 1254. 0818.0038.4180.0A90.0000.1255  ;2475  R[2]_K[.80],D_K[.80]
U 1255. 0019.0030.4580.0A88.0000.1256  ;2476  ALU_D[OR]K[.8000],R[1]_ALU
U 1256. 0818.0038.7980.0800.0000.1048  ;2477  D_K[.30]  ; GENERATE THE ADDRESSES
U 1048. 0819.0031.D580.0800.0000.11A0  ;2478 =0 ALU_D[OR]K[.6],D_ALU,CALL,J/GENTRA ; OF THE FPA UCODE
U 1049. 0001.003C.0180.0AC0.0000.1257  ;2479  R[8]_D
U 1257. 0818.0038.3580.0800.0000.1258  ;2480  D_K[.50]
U 1258. 0018.0038.65C0.0800.0000.104A  ;2481  Q_K[.10]
U 104A. 081D.0015.0180.0800.0000.11A0  ;2482 =0 ALU_D+Q,D_ALU,CALL,J/GENTRA
U 104B. 0001.003C.0180.0AC8.0000.1259  ;2483  R[9]_D
U 1259. 0818.0038.4180.0800.0000.104C  ;2484  D_K[.80]  ; GENERATE ADDRESS 8C SO THAT
U 104C. 0819.0031.8580.0800.0000.11A0  ;2485 =0 ALU_D[OR]K[.C],D_ALU,CALL,J/GENTRA ; THIS TEST CAN TRAP THERE.
U 104D. 0001.003C.0180.0AD0.0000.125A  ;2486  R[0A]_D
U 125A. 0818.0038.7580.0AD8.0000.125B  ;2487  D_K[.20],R[0B]_K[.20]
U 125B. 0819.0014.1180.0AE0.0000.125C  ;2488  ALU_D+K[.4],D_ALU,R[0C]_ALU
U 125C. 0819.0014.1180.0AE8.0000.125D  ;2489  ALU_D+K[.4],D_ALU,R[0D]_ALU
U 125D. 0819.0014.1180.0AF0.0000.1050  ;2490  ALU_D+K[.4],D_ALU,R[0E]_ALU
        ;2491 ;////////////////////////////////////////////////////////////
        ;2492 ;+
        ;2493 ; SUBTEST 1
        ;2494 ;
        ;2495 ; LA=39
        ;2496 ; SA=0
        ;2497 ; LB=1
        ;2498 ; SB=0
        ;2499 ; IB=ADDD
        ;2500 ; FADC<3>=1
        ;2501 ; EXPECTED NAD=6
        ;2502 ;-
U 1050. 0000.003D.0180.0800.0000.1196  ;2503 =0 SUBTEST
U 1051. 0000.003C.0180.0800.0000.1052  ;2504  NOP
U 1052. 0000.003D.0180.0800.0000.1191  ;2505 =0 ERLOOP
U 1053. 0000.003C.0180.0800.0000.1054  ;2506  NOP
U 1054. 0000.003D.0180.0800.0000.119B  ;2507 =0 CALL,J/FPINIT
U 1055. 0000.003C.0180.0A58.0200.1058  ;2508  VA_R[0B]  ; LOAD ADDD INTO IB.
U 1058. 0000.003D.0180.0800.0000.11A6  ;2509 =0 CALL,J/LOADIB
U 1059. 0800.003C.0180.0A40.0000.125E  ;2510  D_R[8]
U 125E. 0000.003C.5980.3C00.0000.125F  ;2511  ID[ACC.2]_D
U 125F. 0000.00FC.0380.0800.0000.1260  ;2512  TRAP.FPA  ; TRAP TO FPA 36
U 1260. 0800.003C.0180.0A50.0000.1261  ;2513  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 1261. 0000.003C.5980.3C00.0000.1262  ;2514  ID[ACC.2]_D  ; AT FPA 36
U 1262. 0000.003C.0180.0A18.0000.1263  ;2515  LAB_R[3]  ; AT FPA 36
U 1263. 0000.007C.0180.0A18.0000.1264  ;2516  LAB_R[3],CPSYNC  ; AT FPA 36
U 1264. 0000.003C.0180.0A10.0000.1265  ;2517  LAB_R[2]  ; AT FPA 37
U 1265. 0000.007C.0180.0A10.0000.1266  ;2518  LAB_R[2],CPSYNC  ; AT FPA 37
U 1266. 0000.00FC.0380.0800.0000.1267  ;2519  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 1267. 0000.003C.0180.0800.0000.1268  ;2520  NOP  ; AT FPA 8C
U 1268. 0000.003C.59F0.2C00.0000.1269  ;2521  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 1269. 0019.2034.81C0.09E8.0000.126A  ;2522  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
```

```
U 126A. 0818.0038.D580.09F0.0000.126B  ;2523  D_K[.6],RC[0E]_K[.6]
U 126B. 001D.0000.0180.0800.0010.126C  ;2524  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 126C. 0000.013C.0180.0800.0000.105C  ;2525  Z?
U 105C. 0000.003D.0180.0800.0000.1192  ;2526 =0 ERROR1   ; BEN/5 FAILED.
U 105D. 0000.003C.0180.0800.0000.1060  ;2527  NOP
          ;2528 ;////////////////////////////////////////////////////////////
          ;2529 ;*
          ;2530 ; SUBTEST 2
          ;2531 ;
          ;2532 ; LA=1
          ;2533 ; SA=0
          ;2534 ; LB=39
          ;2535 ; SB=0
          ;2536 ; IB=ADDD
          ;2537 ; FADC<3>=1
          ;2538 ; EXPECTED NAD=6
          ;2539 ;-
U 1060. 0000.003D.0180.0800.0000.1196  ;2540 =0 SUBTEST
U 1061. 0000.003C.0180.0800.0000.1062  ;2541  NOP
U 1062. 0000.003D.0180.0800.0000.1191  ;2542 =0 ERLOOP
U 1063. 0000.003C.0180.0800.0000.1064  ;2543  NOP
U 1064. 0000.003D.0180.0800.0000.119B  ;2544 =0 CALL,J/FPINIT
U 1065. 0000.003C.0180.0A58.0200.1068  ;2545  VA_R[0B]  ; LOAD ADDD INTO IB.
U 1068. 0000.003D.0180.0800.0000.11A6  ;2546 =0 CALL,J/LOADIB
U 1069. 0800.003C.0180.0A40.0000.126D  ;2547  D_R[8]
U 126D. 0000.003C.5980.3C00.0000.126E  ;2548  ID[ACC.2]_D
U 126E. 0000.00FC.0380.0800.0000.126F  ;2549  TRAP.FPA  ; TRAP TO FPA 36
U 126F. 0800.003C.0180.0A50.0000.1270  ;2550  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 1270. 0000.003C.5980.3C00.0000.1271  ;2551  ID[ACC.2]_D  ; AT FPA 36
U 1271. 0000.003C.0180.0A10.0000.1272  ;2552  LAB_R[2]  ; AT FPA 36
U 1272. 0000.007C.0180.0A10.0000.1273  ;2553  LAB_R[2],CPSYNC  ; AT FPA 36
U 1273. 0000.003C.0180.0A18.0000.1274  ;2554  LAB_R[3]  ; AT FPA 37
U 1274. 0000.007C.0180.0A18.0000.1275  ;2555  LAB_R[3],CPSYNC  ; AT FPA 37
U 1275. 0000.00FC.0380.0800.0000.1276  ;2556  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 1276. 0000.003C.0180.0800.0000.1277  ;2557  NOP  ; AT FPA 8C
U 1277. 0000.003C.59F0.2C00.0000.1278  ;2558  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 1278. 0019.2034.81C0.09E8.0000.1279  ;2559  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1279. 0818.0038.D580.09F0.0000.127A  ;2560  D_K[.6],RC[0E]_K[.6]
U 127A. 001D.0000.0180.0800.0010.127B  ;2561  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 127B. 0000.013C.0180.0800.0000.106A  ;2562  Z?
U 106A. 0000.003D.0180.0800.0000.1192  ;2563 =0 ERROR1   ; BEN/5 FAILED.
U 106B. 0000.003C.0180.0800.0000.106C  ;2564  NOP
          ;2565 ;////////////////////////////////////////////////////////////
          ;2566 ;*
          ;2567 ; SUBTEST 3
          ;2568 ;
          ;2569 ; LA=39
          ;2570 ; SA=1
          ;2571 ; LB=1
          ;2572 ; SB=0
          ;2573 ; IB=ADDD
          ;2574 ; FADC<3>=1
          ;2575 ; EXPECTED NAD=7
          ;2576 ;-
U 106C. 0000.003D.0180.0800.0000.1196  ;2577 =0 SUBTEST
```

```
U 106D, 0000.003C.0180.0800.0000.1070  ;2578  NOP
U 1070, 0000.003D.0180.0800.0000.1191  ;2579 =0 ERLOOP
U 1071, 0000.003C.0180.0800.0000.1072  ;2580  NOP
U 1072, 0000.003D.0180.0800.0000.119B  ;2581 =0 CALL,J/FPINIT
U 1073, 0000.003C.0180.0A58.0200.1074  ;2582  VA_R[0B]  ; LOAD ADDD INTO IB.
U 1074, 0000.003D.0180.0800.0000.11A6  ;2583 =0 CALL,J/LOADIB
U 1075, 0800.003C.0180.0A40.0000.127C  ;2584  D_R[8]
U 127C, 0000.003C.5980.3C00.0000.127D  ;2585  ID[ACC.2]_D
U 127D, 0000.00FC.0380.0800.0000.127E  ;2586  TRAP.FPA  ; TRAP TO FPA 36
U 127E, 0800.003C.0180.0A50.0000.127F  ;2587  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 127F, 0000.003C.5980.3C00.0000.1280  ;2588  ID[ACC.2]_D  ; AT FPA 36
U 1280, 0000.003C.0180.0A20.0000.1281  ;2589  LAB_R[4]  ; AT FPA 36
U 1281, 0000.007C.0180.0A20.0000.1282  ;2590  LAB_R[4],CPSYNC  ; AT FPA 36
U 1282, 0000.003C.0180.0A10.0000.1283  ;2591  LAB_R[2]  ; AT FPA 37
U 1283, 0000.007C.0180.0A10.0000.1284  ;2592  LAB_R[2],CPSYNC  ; AT FPA 37
U 1284, 0000.00FC.0380.0800.0000.1285  ;2593  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 1285, 0000.003C.0180.0800.0000.1286  ;2594  NOP  ; AT FPA 8C
U 1286, 0000.003C.59F0.2C00.0000.1287  ;2595  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 1287, 0019.2034.81C0.09E8.0000.1288  ;2596  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1288, 0818.0038.5D80.09F0.0000.1289  ;2597  D_K[.7],RC[0E]_K[.7]
U 1289, 001D.0000.0180.0800.0010.128A  ;2598  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 128A, 0000.013C.0180.0800.0000.1076  ;2599  Z?
U 1076, 0000.003D.0180.0800.0000.1192  ;2600 =0 ERROR1  ; BEN/5 FAILED.
U 1077, 0000.003C.0180.0800.0000.1078  ;2601  NOP
          ;2602 ;////////////////////////////////////////////////////////////
          ;2603 ;*
          ;2604 ; SUBTEST 4
          ;2605 ;
          ;2606 ; LA=1
          ;2607 ; SA=0
          ;2608 ; LB=39
          ;2609 ; SB=1
          ;2610 ; IB=ADDD
          ;2611 ; FADC<3>=1
          ;2612 ; EXPECTED NAD=7
          ;2613 ;-
U 1078, 0000.003D.0180.0800.0000.1196  ;2614 =0 SUBTEST
U 1079, 0000.003C.0180.0800.0000.107A  ;2615  NOP
U 107A, 0000.003D.0180.0800.0000.1191  ;2616 =0 ERLOOP
U 107B, 0000.003C.0180.0800.0000.107C  ;2617  NOP
U 107C, 0000.003D.0180.0800.0000.119B  ;2618 =0 CALL,J/FPINIT
U 107D, 0000.003C.0180.0A58.0200.107E  ;2619  VA_R[0B]  ; LOAD ADDD INTO IB.
U 107E, 0000.003D.0180.0800.0000.11A6  ;2620 =0 CALL,J/LOADIB
U 107F, 0800.003C.0180.0A40.0000.128B  ;2621  D_R[8]
U 128B, 0000.003C.5980.3C00.0000.128C  ;2622  ID[ACC.2]_D
U 128C, 0000.00FC.0380.0800.0000.128D  ;2623  TRAP.FPA  ; TRAP TO FPA 36
U 128D, 0800.003C.0180.0A50.0000.128E  ;2624  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 128E, 0000.003C.5980.3C00.0000.128F  ;2625  ID[ACC.2]_D  ; AT FPA 36
U 128F, 0000.003C.0180.0A10.0000.1290  ;2626  LAB_R[2]  ; AT FPA 36
U 1290, 0000.007C.0180.0A10.0000.1291  ;2627  LAB_R[2],CPSYNC  ; AT FPA 36
U 1291, 0000.003C.0180.0A20.0000.1292  ;2628  LAB_R[4]  ; AT FPA 37
U 1292, 0000.007C.0180.0A20.0000.1293  ;2629  LAB_R[4],CPSYNC  ; AT FPA 37
U 1293, 0000.00FC.0380.0800.0000.1294  ;2630  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 1294, 0000.003C.0180.0800.0000.1295  ;2631  NOP  ; AT FPA 8C
U 1295, 0000.003C.59F0.2C00.0000.1296  ;2632  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
```

```
U 1296. 0019,2034,81C0,09E8,0000,1297  ;2633  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1297. 0818,0038,5D80,09F0,0000,1298  ;2634  D_K[.7],RC[0E]_K[.7]
U 1298. 001D,0000,0180,0800,0010,1299  ;2635  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 1299. 0000,013C,0180,0800,0000,1080  ;2636  Z?
U 1080. 0000,003D,0180,0800,0000,1192  ;2637 =0 ERROR1   ; BEN/S FAILED.
U 1081. 0000,003C,0180,0800,0000,1082  ;2638  NOP
        ;2639 ;////////////////////////////////////////////////////////////
        ;2640 ;*
        ;2641 ; SUBTEST 5
        ;2642 ;
        ;2643 ; LA=39
        ;2644 ; SA=1
        ;2645 ; LB=1
        ;2646 ; SB=0
        ;2647 ; IB=ADDF
        ;2648 ; FADC<3>=1
        ;2649 ; EXPECTED NAD=6
        ;2650 ;-
U 1082. 0000,003D,0180,0800,0000,1196  ;2651 =0 SUBTEST
U 1083. 0000,003C,0180,0800,0000,1084  ;2652  NOP
U 1084. 0000,003D,0180,0800,0000,1191  ;2653 =0 ERLOOP
U 1085. 0000,003C,0180,0800,0000,1086  ;2654  NOP
U 1086. 0000,003D,0180,0800,0000,119B  ;2655 =0 CALL,J/FPINIT
U 1087. 0000,003C,0180,0A60,0200,1088  ;2656  VA_R[0C]  ; LOAD ADDF INTO IB.
U 1088. 0000,003D,0180,0800,0000,11A6  ;2657 =0 CALL,J/LOADIB
U 1089. 0800,003C,0180,0A40,0000,129A  ;2658  D_R[8]
U 129A. 0000,003C,5980,3C00,0000,129B  ;2659  ID[ACC.2]_D
U 129B. 0000,00FC,0380,0800,0000,129C  ;2660  TRAP.FPA  ; TRAP TO FPA 36
U 129C. 0800,003C,0180,0A50,0000,129D  ;2661  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 129D. 0000,003C,5980,3C00,0000,129E  ;2662  ID[ACC.2]_D  ; AT FPA 36
U 129E. 0000,003C,0180,0A20,0000,129F  ;2663  LAB_R[4]  ; AT FPA 36
U 129F. 0000,007C,0180,0A20,0000,12A0  ;2664  LAB_R[4],CPSYNC  ; AT FPA 36
U 12A0. 0000,003C,0180,0A10,0000,12A1  ;2665  LAB_R[2]  ; AT FPA 37
U 12A1. 0000,007C,0180,0A10,0000,12A2  ;2666  LAB_R[2],CPSYNC  ; AT FPA 37
U 12A2. 0000,00FC,0380,0800,0000,12A3  ;2667  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 12A3. 0000,003C,0180,0800,0000,12A4  ;2668  NOP  ; AT FPA 8C
U 12A4. 0000,003C,59F0,2C00,0000,12A5  ;2669  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 12A5. 0019,2034,81C0,09E8,0000,12A6  ;2670  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12A6. 0818,0038,D580,09F0,0000,12A7  ;2671  D_K[.6],RC[0E]_K[.6]
U 12A7. 001D,0000,0180,0800,0010,12A8  ;2672  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12A8. 0000,013C,0180,0800,0000,108A  ;2673  Z?
U 108A. 0000,003D,0180,0800,0000,1192  ;2674 =0 ERROR1   ; BEN/S FAILED.
U 108B. 0000,003C,0180,0800,0000,108C  ;2675  NOP
        ;2676 ;////////////////////////////////////////////////////////////
        ;2677 ;*
        ;2678 ; SUBTEST 6
        ;2679 ;
        ;2680 ; LA=1
        ;2681 ; SA=0
        ;2682 ; LB=39
        ;2683 ; SB=1
        ;2684 ; IB=ADDF
        ;2685 ; FADC<3>=1
        ;2686 ; EXPECTED NAD=6
        ;2687 ;-
```

```
U 108C, 0000,003D,0180,0800,0000,1196  ;2688 =0 SUBTEST
U 108D, 0000,003C,0180,0800,0000,108E  ;2689  NOP
U 108E, 0000,003D,0180,0800,0000,1191  ;2690 =0 ERLOOP
U 108F, 0000,003C,0180,0800,0000,1090  ;2691  NOP
U 1090, 0000,003D,0180,0800,0000,119B  ;2692 =0 CALL,J/FPINIT
U 1091, 0000,003C,0180,0A60,0200,1092  ;2693  VA_R[0C]  ; LOAD ADDF INTO IB.
U 1092, 0000,003D,0180,0800,0000,11A6  ;2694 =0 CALL,J/LOADIB
U 1093, 0800,003C,0180,0A40,0000,12A9  ;2695  D_R[8]
U 12A9, 0000,003C,5980,3C00,0000,12AB  ;2696  ID[ACC.2]_D
U 12AB, 0000,00FC,0380,0800,0000,12AC  ;2697  TRAP.FPA  ; TRAP TO FPA 36
U 12AC, 0800,003C,0180,0A50,0000,12AD  ;2698  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 12AD, 0000,003C,5980,3C00,0000,12AE  ;2699  ID[ACC.2]_D  ; AT FPA 36
U 12AE, 0000,003C,0180,0A10,0000,12AF  ;2700  LAB_R[2]  ; AT FPA 36
U 12AF, 0000,007C,0180,0A10,0000,12B0  ;2701  LAB_R[2],CPSYNC  ; AT FPA 36
U 12B0, 0000,003C,0180,0A20,0000,12B1  ;2702  LAB_R[4]  ; AT FPA 37
U 12B1, 0000,007C,0180,0A20,0000,12B2  ;2703  LAB_R[4],CPSYNC  ; AT FPA 37
U 12B2, 0000,00FC,0380,0800,0000,12B3  ;2704  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 12B3, 0000,003C,0180,0800,0000,12B4  ;2705  NOP  ; AT FPA 8C
U 12B4, 0000,003C,59F0,2C00,0000,12B5  ;2706  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 12B5, 0019,2034,81C0,09E8,0000,12B6  ;2707  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12B6, 0818,0038,D580,09F0,0000,12B7  ;2708  D_K[.6],RC[0E]_K[.6]
U 12B7, 001D,0000,0180,0800,0010,12B8  ;2709  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12B8, 0000,013C,0180,0800,0000,1094  ;2710  Z?
U 1094, 0000,003D,0180,0800,0000,1192  ;2711 =0 ERROR1  ; BEN/5 FAILED.
U 1095, 0000,003C,0180,0800,0000,1096  ;2712  NOP
          ;2713 ;////////////////////////////////////////////////////////////
          ;2714 ;*
          ;2715 ; SUBTEST 7
          ;2716 ;
          ;2717 ; LA=39
          ;2718 ; SA=1
          ;2719 ; LB=1
          ;2720 ; SB=0
          ;2721 ; IB=ADDD
          ;2722 ; FADC<3>=0
          ;2723 ; EXPECTED NAD=6
          ;2724 ;-
U 1096, 0000,003D,0180,0800,0000,1196  ;2725 =0 SUBTEST
U 1097, 0000,003C,0180,0800,0000,1098  ;2726  NOP
U 1098, 0000,003D,0180,0800,0000,1191  ;2727 =0 ERLOOP
U 1099, 0000,003C,0180,0800,0000,109A  ;2728  NOP
U 109A, 0000,003D,0180,0800,0000,119B  ;2729 =0 CALL,J/FPINIT
U 109B, 0000,003C,0180,0A58,0200,109C  ;2730  VA_R[0B]  ; LOAD ADDD INTO IB.
U 109C, 0000,003D,0180,0800,0000,11A6  ;2731 =0 CALL,J/LOADIB
U 109D, 0800,003C,0180,0A48,0000,12B9  ;2732  D_R[9]
U 12B9, 0000,003C,5980,3C00,0000,12BA  ;2733  ID[ACC.2]_D
U 12BA, 0000,00FC,0380,0800,0000,12BB  ;2734  TRAP.FPA  ; TRAP TO FPA
U 12BB, 0000,003C,0180,0A20,0000,12BC  ;2735  LAB_R[4]  ; AT FPA 60
U 12BC, 0000,007C,0180,0A20,0000,12BD  ;2736  LAB_R[4],CPSYNC  ; AT FPA 60
U 12BD, 0000,003C,0180,0A10,0000,12BE  ;2737  LAB_R[2]  ; AT FPA 61
U 12BE, 0000,007C,0180,0A10,0000,12BF  ;2738  LAB_R[2],CPSYNC  ; AT FPA 61
U 12BF, 0000,003C,0180,0800,0000,12C0  ;2739  NOP  ; AT FPA 8D
U 12C0, 0000,003C,59F0,2C00,0000,12C1  ;2740  Q_ID[ACC.2]  ; AT FPA 62, GET NAD
U 12C1, 0019,2034,81C0,09E8,0000,12C2  ;2741  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12C2, 0818,0038,D580,09F0,0000,12C3  ;2742  D_K[.6],RC[0E]_K[.6]
```

```
U 12C3, 001D,0000,0180,0800,0010,12C4  ;2743  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12C4, 0000,013C,0180,0800,0000,109E  ;2744  Z?
U 109E, 0000,003D,0180,0800,0000,1192  ;2745 =0 ERROR1    ; BEN/5 FAILED.
U 109F, 0000,003C,0180,0800,0000,10A0  ;2746  NOP
            ;2747 ;////////////////////////////////////////////////////////////
            ;2748 ;*
            ;2749 ; SUBTEST 8
            ;2750 ;
            ;2751 ; LA=1
            ;2752 ; SA=0
            ;2753 ; LB=39
            ;2754 ; SB=1
            ;2755 ; IB=ADDD
            ;2756 ; FADC<3>=0
            ;2757 ; EXPECTED NAD=6
            ;2758 ;-
U 10A0, 0000,003D,0180,0800,0000,1196  ;2759 =0 SUBTEST
U 10A1, 0000,003C,0180,0800,0000,10A2  ;2760  NOP
U 10A2, 0000,003D,0180,0800,0000,1191  ;2761 =0 ERLOOP
U 10A3, 0000,003C,0180,0800,0000,10A4  ;2762  NOP
U 10A4, 0000,003D,0180,0800,0000,119B  ;2763 =0 CALL,J/FPINIT
U 10A5, 0000,003C,0180,0A58,0200,10A6  ;2764  VA_R[0B]  ; LOAD ADDD INTO IB.
U 10A6, 0000,003D,0180,0800,0000,11A6  ;2765 =0 CALL,J/LOADIB
U 10A7, 0800,003C,0180,0A48,0000,12C5  ;2766  D_R[9]
U 12C5, 0000,003C,5980,3C00,0000,12C6  ;2767  ID[ACC.2]_D
U 12C6, 0000,00FC,0380,0800,0000,12C7  ;2768  TRAP.FPA  ; TRAP TO FPA 60
U 12C7, 0000,003C,0180,0A10,0000,12C8  ;2769  LAB_R[2]  ; AT FPA 60
U 12C8, 0000,007C,0180,0A10,0000,12C9  ;2770  LAB_R[2],CPSYNC  ; AT FPA 60
U 12C9, 0000,003C,0180,0A20,0000,12CA  ;2771  LAB_R[4]  ; AT FPA 61
U 12CA, 0000,007C,0180,0A20,0000,12CB  ;2772  LAB_R[4],CPSYNC  ; AT FPA 61
U 12CB, 0000,003C,0180,0800,0000,12CC  ;2773  NOP  ; AT FPA 8D
U 12CC, 0000,003C,59F0,2C00,0000,12CD  ;2774  Q_ID[ACC.2]  ; AT FPA 62, GET NAD
U 12CD, 0019,2034,81C0,09E8,0000,12CE  ;2775  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12CE, 0818,0038,D580,09F0,0000,12CF  ;2776  D_K[.6],RC[0E]_K[.6]
U 12CF, 001D,0000,0180,0800,0010,12D0  ;2777  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12D0, 0000,013C,0180,0800,0000,10A8  ;2778  Z?
U 10A8, 0000,003D,0180,0800,0000,1192  ;2779 =0 ERROR1    ; BEN/5 FAILED.
U 10A9, 0000,003C,0180,0800,0000,10AA  ;2780  NOP
            ;2781 ;////////////////////////////////////////////////////////////
            ;2782 ;*
            ;2783 ; SUBTEST 9
            ;2784 ;
            ;2785 ; LA=39
            ;2786 ; SA=1
            ;2787 ; LB=39
            ;2788 ; SB=0
            ;2789 ; IB=POLYD
            ;2790 ; FADC<3>=1
            ;2791 ; EXPECTED NAD=7
            ;2792 ;-
U 10AA, 0000,003D,0180,0800,0000,1196  ;2793 =0 SUBTEST
U 10AB, 0000,003C,0180,0800,0000,10AC  ;2794  NOP
U 10AC, 0000,003D,0180,0800,0000,1191  ;2795 =0 ERLOOP
U 10AD, 0000,003C,0180,0800,0000,10AE  ;2796  NOP
U 10AE, 0000,003D,0180,0800,0000,119B  ;2797 =0 CALL,J/FPINIT
```

```
U 10AF, 0000,003C,0180,0A68,0200,10B0   ;2798  VA_R[0D]  ; LOAD POLYD INTO IB.
U 10B0, 0000,003D,0180,0800,0000,11A6   ;2799 =0 CALL,J/LOADIB
U 10B1, 0800,003C,0180,0A40,0000,12D1   ;2800  D_R[8]
U 12D1, 0000,003C,5980,3C00,0000,12D2   ;2801  ID[ACC.2]_D
U 12D2, 0000,00FC,0380,0800,0000,12D3   ;2802  TRAP.FPA  ; TRAP TO FPA 36
U 12D3, 0800,003C,0180,0A50,0000,12D4   ;2803  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 12D4, 0000,003C,5980,3C00,0000,12D5   ;2804  ID[ACC.2]_D  ; AT FPA 36
U 12D5, 0000,003C,0180,0A20,0000,12D6   ;2805  LAB_R[4]  ; AT FPA 36
U 12D6, 0000,007C,0180,0A20,0000,12D7   ;2806  LAB_R[4],CPSYNC  ; AT FPA 36
U 12D7, 0000,003C,0180,0A18,0000,12D8   ;2807  LAB_R[3]  ; AT FPA 37
U 12D8, 0000,007C,0180,0A18,0000,12D9   ;2808  LAB_R[3],CPSYNC  ; AT FPA 37
U 12D9, 0000,00FC,0380,0800,0000,12DA   ;2809  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 12DA, 0000,003C,0180,0800,0000,12DB   ;2810  NOP  ; AT FPA 8C
U 12DB, 0000,003C,59F0,2C00,0000,12DC   ;2811  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 12DC, 0019,2034,81C0,09E8,0000,12DD   ;2812  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12DD, 0818,0038,5D80,09F0,0000,12DE   ;2813  D_K[.7],RC[0E]_K[.7]
U 12DE, 001D,0000,0180,0800,0010,12DF   ;2814  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12DF, 0000,013C,0180,0800,0000,10B2   ;2815  Z?
U 10B2, 0000,003D,0180,0800,0000,1192   ;2816 =0 ERROR1  ; BEN/S FAILED.
U 10B3, 0000,003C,0180,0800,0000,10B4   ;2817  NOP
          ;2818 ;////////////////////////////////////////////////////////////
          ;2819 ;*
          ;2820 ; SUBTEST A
          ;2821 ;
          ;2822 ; LA=39
          ;2823 ; SA=0
          ;2824 ; LB=39
          ;2825 ; SB=0
          ;2826 ; IB=POLYD
          ;2827 ; FADC<3>=1
          ;2828 ; EXPECTED NAD=6
          ;2829 ;-
U 10B4, 0000,003D,0180,0800,0000,1196   ;2830 =0 SUBTEST
U 10B5, 0000,003C,0180,0800,0000,10B6   ;2831  NOP
U 10B6, 0000,003D,0180,0800,0000,1191   ;2832 =0 ERLOOP
U 10B7, 0000,003C,0180,0800,0000,10B8   ;2833  NOP
U 10B8, 0000,003D,0180,0800,0000,119B   ;2834 =0 CALL,J/FPINIT
U 10B9, 0000,003C,0180,0A68,0200,10BA   ;2835  VA_R[0D]  ; LOAD POLYD INTO IB.
U 10BA, 0000,003D,0180,0800,0000,11A6   ;2836 =0 CALL,J/LOADIB
U 10BB, 0800,003C,0180,0A40,0000,12E0   ;2837  D_R[8]
U 12E0, 0000,003C,5980,3C00,0000,12E1   ;2838  ID[ACC.2]_D
U 12E1, 0000,00FC,0380,0800,0000,12E2   ;2839  TRAP.FPA  ; TRAP TO FPA 36
U 12E2, 0800,003C,0180,0A50,0000,12E3   ;2840  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 12E3, 0000,003C,5980,3C00,0000,12E4   ;2841  ID[ACC.2]_D  ; AT FPA 36
U 12E4, 0000,003C,0180,0A18,0000,12E5   ;2842  LAB_R[3]  ; AT FPA 36
U 12E5, 0000,007C,0180,0A18,0000,12E6   ;2843  LAB_R[3],CPSYNC  ; AT FPA 36
U 12E6, 0000,003C,0180,0A18,0000,12E7   ;2844  LAB_R[3]  ; AT FPA 37
U 12E7, 0000,007C,0180,0A18,0000,12E8   ;2845  LAB_R[3],CPSYNC  ; AT FPA 37
U 12E8, 0000,00FC,0380,0800,0000,12E9   ;2846  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 12E9, 0000,003C,0180,0800,0000,12EA   ;2847  NOP  ; AT FPA 8C
U 12EA, 0000,003C,59F0,2C00,0000,12EB   ;2848  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 12EB, 0019,2034,81C0,09E8,0000,12EC   ;2849  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12EC, 0818,0038,D580,09F0,0000,12ED   ;2850  D_K[.6],RC[0E]_K[.6]
U 12ED, 001D,0000,0180,0800,0010,12EE   ;2851  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12EE, 0000,013C,0180,0800,0000,10BC   ;2852  Z?
```

```
U 10BC, 0000,003D,0180,0800,0000,1192  ;2853 =0 ERROR1   ; BEN/5 FAILED.
U 10BD, 0000,003C,0180,0800,0000,10BE  ;2854  NOP
          ;2855 ;////////////////////////////////////////////////////////////
          ;2856 ;*
          ;2857 ; SUBTEST B
          ;2858 ;
          ;2859 ; LA=39
          ;2860 ; SA=1
          ;2861 ; LB=39
          ;2862 ; SB=0
          ;2863 ; IB=POLYF
          ;2864 ; FADC<3>=1
          ;2865 ; EXPECTED NAD=6
          ;2866 ;-
U 10BE, 0000,003D,0180,0800,0000,1196  ;2867 =0 SUBTEST
U 10BF, 0000,003C,0180,0800,0000,10C0  ;2868  NOP
U 10C0, 0000,003D,0180,0800,0000,1191  ;2869 =0 ERLOOP
U 10C1, 0000,003C,0180,0800,0000,10C2  ;2870  NOP
U 10C2, 0000,003D,0180,0800,0000,119B  ;2871 =0 CALL,J/FPINIT
U 10C3, 0000,003C,0180,0A70,0200,10C4  ;2872  VA_R[0E]  ; LOAD POLYF INTO IB.
U 10C4, 0000,003D,0180,0800,0000,11A6  ;2873 =0 CALL,J/LOADIB
U 10C5, 0800,003C,0180,0A40,0000,12EF  ;2874  D_R[8]
U 12EF, 0000,003C,5980,3C00,0000,12F0  ;2875  ID[ACC.2]_D
U 12F0, 0000,00FC,0380,0800,0000,12F1  ;2876  TRAP.FPA  ; TRAP TO FPA 36
U 12F1, 0800,003C,0180,0A50,0000,12F2  ;2877  D_R[0A]  ; AT FPA 36, SET UP FOR TRAP TO 8C
U 12F2, 0000,003C,5980,3C00,0000,12F3  ;2878  ID[ACC.2]_D  ; AT FPA 36
U 12F3, 0000,003C,0180,0A20,0000,12F4  ;2879  LAB_R[4]  ; AT FPA 36
U 12F4, 0000,007C,0180,0A20,0000,12F5  ;2880  LAB_R[4],CPSYNC  ; AT FPA 36
U 12F5, 0000,003C,0180,0A18,0000,12F6  ;2881  LAB_R[3]  ; AT FPA 37
U 12F6, 0000,007C,0180,0A18,0000,12F7  ;2882  LAB_R[3],CPSYNC  ; AT FPA 37
U 12F7, 0000,00FC,0380,0800,0000,12F8  ;2883  TRAP.FPA  ; AT FPA 8C, TRAP TO STAY AT 8C
U 12F8, 0000,003C,0180,0800,0000,12F9  ;2884  NOP  ; AT FPA 8C
U 12F9, 0000,003C,59F0,2C00,0000,12FA  ;2885  Q_ID[ACC.2]  ; AT FPA 38, GET NAD
U 12FA, 0019,2034,81C0,09E8,0000,12FB  ;2886  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 12FB, 0818,0038,D580,09F0,0000,12FC  ;2887  D_K[.6],RC[0E]_K[.6]
U 12FC, 001D,0000,0180,0800,0010,12FD  ;2888  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 12FD, 0000,013C,0180,0800,0000,10C6  ;2889  Z?
U 10C6, 0000,003D,0180,0800,0000,1192  ;2890 =0 ERROR1   ; BEN/5 FAILED.
U 10C7, 0000,003C,0180,0800,0000,10C8  ;2891  NOP
          ;2892 ;////////////////////////////////////////////////////////////
          ;2893 ;*
          ;2894 ; SUBTEST C
          ;2895 ;
          ;2896 ; LA=39
          ;2897 ; SA=0
          ;2898 ; LB=39
          ;2899 ; SB=1
          ;2900 ; IB=POLYD
          ;2901 ; FADC<3>=0
          ;2902 ; EXPECTED NAD=6
          ;2903 ;-
U 10C8, 0000,003D,0180,0800,0000,1196  ;2904 =0 SUBTEST
U 10C9, 0000,003C,0180,0800,0000,10CA  ;2905  NOP
U 10CA, 0000,003D,0180,0800,0000,1191  ;2906 =0 ERLOOP
U 10CB, 0000,003C,0180,0800,0000,10CC  ;2907  NOP
```

```
U 10CC. 0000.003D.0180.0800.0000.119B  ;2908 =0 CALL.J/FPINIT
U 10CD. 0000.003C.0180.0A68.0200.10CE  ;2909  VA_R[0D]  ; LOAD POLYD INTO IB.
U 10CE. 0000.003D.0180.0800.0000.11A6  ;2910 =0 CALL.J/LOADIB
U 10CF. 0800.003C.0180.0A48.0000.12FE  ;2911  D_R[9]
U 12FE. 0000.003C.5980.3C00.0000.12FF  ;2912  ID[ACC.2]_D
U 12FF. 0000.00FC.0380.0800.0000.1300  ;2913  TRAP.FPA  ; TRAP TO FPA 60
U 1300. 0000.003C.0180.0A18.0000.1301  ;2914  LAB_R[3]  ; AT FPA 60
U 1301. 0000.007C.0180.0A18.0000.1302  ;2915  LAB_R[3],CPSYNC  ; AT FPA 60
U 1302. 0000.003C.0180.0A20.0000.1303  ;2916  LAB_R[4]  ; AT FPA 61
U 1303. 0000.007C.0180.0A20.0000.1304  ;2917  LAB_R[4],CPSYNC  ; AT FPA 61
U 1304. 0000.003C.0180.0800.0000.1305  ;2918  NOP  ; AT FPA 8D
U 1305. 0000.003C.59F0.2C00.0000.1306  ;2919  Q_ID[ACC.2]  ; AT FPA 62, GET MAD
U 1306. 0019.2034.81C0.09E8.0000.1307  ;2920  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1307. 0818.0038.D580.09F0.0000.1308  ;2921  D_K[.6],RC[0E]_K[.6]
U 1308. 001D.0000.0180.0800.0010.1309  ;2922  ALU_D-Q,CLK.UBCC ; CHECK RESULT.
U 1309. 0000.013C.0180.0800.0000.10D0  ;2923  Z?
U 10D0. 0000.003D.0180.0800.0000.1192  ;2924 =0 ERROR1  ; BEN/S FAILED.
U 10D1. 0000.003C.0180.0800.0000.10D2  ;2925  NOP
```

```
                ;2926  .PAGE "TEST 270 BEN/5 BIT<1> TEST 1"
                ;2927 ;*****************************************************************
                ;2928 ;**
                ;2929 ; BEN/5 BIT<1> TEST
                ;2930 ;
                ;2931 ; TEST DESCRIPTION
                ;2932 ; THIS IS A TEST OF BEN/5 BIT<1> WHICH IS ASSERTED WHEN THE
                ;2933 ; EXPONENT DIFFERENCE IS 0 OR 1 AND THE FALU FUNCTION IS SUB.
                ;2934 ; ALL COMBINATIONS OF LA AND LB ARE TRIED AND THE BEN/5 EXECUTED
                ;2935 ; AND CHECKED.
                ;2936 ;
                ;2937 ; LOGIC DESCRIPTION
                ;2938 ;
                ;2939 ; FPA UCODE USED
                ;2940 ; LOCATION CONTENTS
                ;2941 ; 65:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,NAD/66,WAIT
                ;2942 ; 66:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,NAD/67,WAIT
                ;2943 ; 67:  BEN/5,NAD/5
                ;2944 ;
                ;2945 ; ERROR DESCRIPTION
                ;2946 ; EXPECTED NAD
                ;2947 ; GOT NAD
                ;2948 ; CONTENTS OF LA <07:00>
                ;2949 ; CONTENTS OF LB <07:00>
                ;2950 ; EXPONENT DIFFERENCE, ABSOLUTE VALUE OF LA-LB, <07:00>
                ;2951 ;
                ;2952 ; SYNC POINT DESCRIPTION
                ;2953 ;
                ;2954 ;--
                ;2955 ;*****************************************************************
                ;2956 =0
U 10D2, 0000,003D,0180,0800,0000,117F   ;2957 B5B1: NEWTST
U 10D3, 0818,0038,1180,0800,0000,130A   ;2958  D_K[.4]
U 130A, 0019,0014,0580,09F8,0000,130B   ;2959  ALU_D+K[.1],RC[0F]_ALU
U 130B, 0018,0038,4980,0A80,0000,130C   ;2960  R[0]_K[.FF]  ; INITIAL LA VALUE
U 130C, 0018,0038,4980,0A88,0000,10D4   ;2961  R[1]_K[.FF]  ; INITIAL LB VALUE
U 10D4, 0000,003D,0180,0800,0000,1191   ;2962 =0 ERLOOP
U 10D5, 0000,003C,0180,0800,0000,10D6   ;2963  NOP
                ;2964 =0
U 10D6, 0000,003D,0180,0800,0000,119B   ;2965 B5B1A: CALL,J/FPINIT
U 10D7, 0018,0038,7580,0800,0200,10D8   ;2966  VA_K[.20]  ; LOAD ADDD INTO IB.
U 10D8, 0000,003D,0180,0800,0000,11A6   ;2967 =0 CALL,J/LOADIB
U 10D9, 0818,0038,E980,0800,0000,130D   ;2968  D_K[.24]  ; GENERATE THE ADDRESS
U 130D, 0819,0030,3180,0800,0000,10DA   ;2969  ALU_D[OR]K[.40],D_ALU ; OF THE FPA UCODE.
U 10DA, 0819,0015,0580,0800,0000,11A0   ;2970 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 10DB, 0000,003C,5980,3C00,0000,130E   ;2971  ID[ACC.2]_D
U 130E, 0000,003C,5DF8,0800,0084,730F   ;2972  SC_K[.7],Q_0  ; GENERATE CURRENT LA AND LB VALUES.
U 130F, 0800,003C,0180,0A00,0000,1310   ;2973  D_R[0]
U 1310, 0001,003C,0180,09E0,0000,1311   ;2974  RC[0C]_D
U 1311, 0D00,003C,0180,0800,0000,1312   ;2975  D_DAL,SC
U 1312, 0019,0030,4580,0A90,0000,1313   ;2976  ALU_D[OR]K[.8000],R[2]_ALU
U 1313, 0800,003C,0180,0A08,0000,1314   ;2977  D_R[1]
U 1314, 0001,003C,0180,09D8,0000,1315   ;2978  RC[0B]_D
U 1315, 0D00,003C,0180,0800,0000,1316   ;2979  D_DAL,SC
U 1316, 0001,003C,0180,0A98,0000,1317   ;2980  R[3]_D
```

```
U 1317. 0800.003C.0180.0A00.0000.1318   ;2981  D_R[0]
U 1318. 0000.003C.01C0.0A08.0000.1319   ;2982  Q_R[1]    ; COMPUTE EXPECTED RESULT FROM
U 1319. 081D.0000.0180.0800.0010.131A   ;2983  ALU_D-Q,D_ALU,CLK.UBCC ; EXPONENT DIFFERENCE.
U 131A. 0000.1B3C.01F8.0800.0000.1067   ;2984  ALU.N?,Q_0
U 1067. 0001.003C.0180.09D0.0000.131B   ;2985 =0111 RC[0A]_D,J/B5B1B
U 106F. 001D.2000.0180.09D0.0000.131B   ;2986  ALU_Q-D,RC[0A]_ALU
U 131B. 0810.0038.0180.0950.0000.131C   ;2987 B5B1B: D_RC[0A]
U 131C. 0019.0024.0580.0800.0010.131D   ;2988  ALU_D[ANDNOT]K[.1],CLK.UBCC
U 131D. 0000.013C.0180.0800.0000.10DC   ;2989  Z?
U 10DC. 0818.0038.1180.0800.0000.131E   ;2990 =0 J/B5B1C,D_K[.4]
U 10DD. 0018.0038.5D80.09F0.0000.131F   ;2991  RC[0E]_K[.7],J/B5B1D
U 131E. 0019.0014.0580.09F0.0000.131F   ;2992 B5B1C: ALU_D+K[.1],RC[0E]_ALU
U 131F. 0000.00FC.0380.0800.0000.1320   ;2993 B5B1D: TRAP.FPA   ; TRAP TO FPA 65
U 1320. 0000.003C.0180.0A10.0000.1321   ;2994  LAB_R[2]   ; AT FPA 65
U 1321. 0000.007C.0180.0A10.0000.1322   ;2995  LAB_R[2],CPSYNC   ; AT FPA 65
U 1322. 0000.003C.0180.0A18.0000.1323   ;2996  LAB_R[3]   ; AT FPA 66
U 1323. 0000.007C.0180.0A18.0000.1324   ;2997  LAB_R[3],CPSYNC   ; AT FPA 66
U 1324. 0000.003C.59F0.2C00.0000.1325   ;2998  Q_ID[ACC.2]   ; AT FPA 67, GET NAD
U 1325. 0019.2034.81C0.09E8.0000.1326   ;2999  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 1326. 0810.0038.0180.0970.0000.1327   ;3000  D_RC[0E]
U 1327. 001D.0000.0180.0800.0010.1328   ;3001  ALU_D-Q,CLK.UBCC ; CHECK THE RESULT.
U 1328. 0000.013C.0180.0800.0000.10DE   ;3002  Z?
U 10DE. 0000.003D.0180.0800.0000.1192   ;3003 =0 ERROR1    ; BEN 5 FAILED
U 10DF. 0000.003C.0180.0A08.0010.1329   ;3004  ALU_R[1],CLK.UBCC ; TRIED ALL LB VALUES?
U 1329. 0000.013C.0180.0800.0000.10E0   ;3005  Z?
U 10E0. 0018.0000.0580.0A88.0000.10D6   ;3006 =0 R[1]_LA-K[.1],J/B5B1A ; IF NOT DECREMENT AND CONTINUE.
U 10E1. 0018.0038.4980.0A88.0000.132A   ;3007  R[1]_K[.FF]
U 132A. 0000.003C.0180.0A00.0010.132B   ;3008  ALU_R[0],CLK.UBCC ; TRIED ALL LA VALUES?
U 132B. 0000.013C.0180.0800.0000.10E2   ;3009  Z?
U 10E2. 0018.0000.0580.0A80.0000.10D6   ;3010 =0 R[0]_LA-K[.1],J/B5B1A ; IF NOT DECREMENT AND CONTINUE.
U 10E3. 0000.003C.0180.0800.0000.10E4   ;3011  NOP
```

```
                ;3012  .PAGE "TEST 271 BEN/5 BIT<1> TEST 2"
                ;3013 ;*****************************************************************
                ;3014 ;*
                ;3015 ; BEN/5 BIT<1> TEST 2"
                ;3016 ;
                ;3017 ; TEST DESCRIPTION
                ;3018 ; THIS TEST EXECUTES A BEN 5 WITH EXPONENT DIFFERENCE 0 AND
                ;3019 ; FALU ADD, THIS SHOULD NOT ASSERT THE BRANCH BIT.
                ;3020 ;
                ;3021 ; LOGIC DESCRIPTION
                ;3022 ;
                ;3023 ; FPA UCODE USED
                ;3024 ; LOCATION CONTENTS
                ;3025 ; 65:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/66
                ;3026 ; 66:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/67
                ;3027 ; 67:  BEN/5,NAD/5
                ;3028 ;
                ;3029 ; ERROR DESCRIPTION
                ;3030 ; EXPECTED NAD
                ;3031 ; GOT NAD
                ;3032 ; CONTENTS OF LA <07:00>
                ;3033 ; CONTENTS OF LB <07:00>
                ;3034 ; EXPONENT DIFFERENCE
                ;3035 ;
                ;3036 ; SYNC POINT DESCRIPTION
                ;3037 ;
                ;3038 ;--
                ;3039 ;*****************************************************************
                ;3040 =0
U 10E4, 0000,003D,0180,0800,0000,117F   ;3041 BSN: NEWTST
U 10E5, 0818,0038,1180,0800,0000,132C   ;3042  D_K[.4]
U 132C, 0019,0014,0580,09F8,0000,132D   ;3043  ALU_D+K[.1],RC[0F]_ALU
U 132D, 0018,0038,4180,0A90,0000,132E   ;3044  R[2]_K[.80]
U 132E, 0019,0014,0580,09F0,0000,132F   ;3045  ALU_D+K[.1],RC[0E]_ALU
U 132F, 0018,0038,0580,09E0,0000,1330   ;3046  RC[0C]_K[.1]
U 1330, 0018,0038,0580,09D8,0000,1331   ;3047  RC[0B]_K[.1]
U 1331, 0003,003C,0180,09D0,0000,10E6   ;3048  RC[0A]_0
U 10E6, 0000,003D,0180,0800,0000,119B   ;3049 =0 CALL,J/FPINIT
U 10E7, 0018,0038,7580,0800,0200,10E8   ;3050  VA_K[.20]
U 10E8, 0000,003D,0180,0800,0000,11A6   ;3051 =0 CALL,J/LOADIB  ; LOAD ADDD INTO IB.
U 10E9, 0818,0038,E980,0800,0000,1332   ;3052  D_K[.24]  ; GENERATE THE ADDRESS OF THE
U 1332, 0819,0030,3180,0800,0000,10EA   ;3053  ALU_D[OR]K[.40],D_ALU ; FPA UCODE.
U 10EA, 0819,0015,0580,0800,0000,11A0   ;3054 =0 ALU_D+K[.1],D_ALU,CALL,J/GENTRA
U 10EB, 0000,003C,5980,3C00,0000,1333   ;3055  ID[ACC.2]_D
U 1333, 0000,00FC,0380,0800,0000,1334   ;3056  TRAP,FPA  ; TRAP TO FPA 65
U 1334, 0000,003C,0180,0A10,0000,1335   ;3057  LAB_R[2]  ; AT FPA 65
U 1335, 0000,007C,0180,0A10,0000,1336   ;3058  LAB_R[2],CPSYNC  ; AT FPA 65
U 1336, 0000,003C,0180,0A10,0000,1337   ;3059  LAB_R[2]  ; AT FPA 66
U 1337, 0000,007C,0180,0A10,0000,1338   ;3060  LAB_R[2],CPSYNC  ; AT FPA 66
U 1338, 0000,003C,59F0,2C00,0000,1339   ;3061  Q_ID[ACC.2]  ; AT FPA 67, GET NAD
U 1339, 0019,2034,81C0,09E8,0000,133A   ;3062  ALU_Q[AND]K[.3FF],Q_ALU,RC[0D]_ALU
U 133A, 0810,0038,0180,0970,0000,133B   ;3063  D_RC[0E]
U 133B, 001D,0000,0180,0800,0010,133C   ;3064  ALU_D-Q,CLK,UBCC ; CHECK THE RESULT.
U 133C, 0000,013C,0180,0800,0000,10EC   ;3065  Z?
U 10EC, 0000,003D,0180,0800,0000,1192   ;3066 =0 ERROR1   ; BEN/5 FAILED.
```

```
U 10ED, 0000,003C,0180,0800,0000,10EE  ;3067  NOP
```

```
            ;3068  .PAGE "TEST 272 FALU A+B TEST 1"
            ;3069 ;*****************************************************************
            ;3070 ;**
            ;3071 ; FALU A+B TEST 1
            ;3072 ;
            ;3073 ; TEST DESCRIPTION
            ;3074 ; THIS TEST PERFORMS FALU ADD'S WITH OPERANDS WHICH
            ;3075 ; TRY EVERY COMBINATION OF INPUT PATTERNS TO THE 5 BIT
            ;3076 ; FALU SLICES:
            ;3077 ;  FALU <59:55>
            ;3078 ;  FALU <51:47>
            ;3079 ;  FALU <43:39>
            ;3080 ;  FALU <35:31>
            ;3081 ;  FALU <27:23>
            ;3082 ;  FALU <19:15>
            ;3083 ;  FALU <11:07>
            ;3084 ;
            ;3085 ; LOGIC DESCRIPTION
            ;3086 ;
            ;3087 ; FPA UCODE USED
            ;3088 ; LOCATION CONTENTS
            ;3089 ; 4F:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/39
            ;3090 ; 39:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/3A
            ;3091 ; 3A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/3B
            ;3092 ; 3B:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/3C
            ;3093 ; 3C:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/3D
            ;3094 ; 3D:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/3E
            ;3095 ; 3E:  FADC/A.B,BSC/FAL.HL,NAD/3F
            ;3096 ; 3F:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
            ;3097 ;
            ;3098 ; ERROR DESCRIPTION
            ;3099 ; EXPECTED RESULT HI, FROM BUS A
            ;3100 ; EXPECTED RESULT LO, FROM BUS A
            ;3101 ; GOT RESULT HI, FROM BUS A
            ;3102 ; GOT RESULT LO, FROM BUS A
            ;3103 ; EXPECTED OUTPUT OF FALU HI
            ;3104 ; EXPECTED OUTPUT OF FALU LO
            ;3105 ; CONTENTS OF AR HI
            ;3106 ; CONTENTS OF AR LO
            ;3107 ; CONTENTS OF BR HI
            ;3108 ; CONTENTS OF BR LO
            ;3109 ;
            ;3110 ; SYNC POINT DESCRIPTION
            ;3111 ;
            ;3112 ;--
            ;3113 ;*****************************************************************
            ;3114 =0
U 10EE, 0000,003D,0180,0800,0000,117F  ;3115 ADDT1: NEWTST
U 10EF, 0018,0038,F580,09F8,0000,133D  ;3116  RC[0F]_K[.A]
U 133D, 0018,0038,5580,0A80,0000,133E  ;3117  R[0]_K[.3F]  ; INITIALIZE SOME COUNTERS.
U 133E, 0018,0038,5580,0A88,0000,133F  ;3118  R[1]_K[.3F]
U 133F, 0003,003C,01F8,0AF8,0000,1340  ;3119  Q_0,R[0F]_0  ; INITIALIZE AR OPERAND
U 1340, 0003,003C,03B0,09B8,0000,1341  ;3120  Q_Q.RIGHT,SI/7,RC[7]_0 ; AND BR OPERAND
U 1341, 0003,003C,01B0,09A8,0000,1342  ;3121  Q_Q.RIGHT,SI/ZERO,RC[5]_0
U 1342, 0001,203C,0180,09C0,0000,1343  ;3122  RC[8]_Q
```

```
U 1343. 0001.203C.0180.09B0.0000.1344  ;3123  RC[6]_Q
U 1344. 0001.203C.0180.0AF0.0000.1345  ;3124  R[0E]_Q
U 1345. 0000.003C.5D80.0800.0084.7346  ;3125  SC_K[.7]  ; SET UP INCREMENT TO OPERANDS.
U 1346. 0803.001C.0180.0800.0084.9347  ;3126  ALU_NOT.MASK,D_ALU,SC_SC+K[.8]
U 1347. 0801.001C.0180.0800.0084.9348  ;3127  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 1348. 0801.001C.0180.0AE0.0084.9349  ;3128  ALU_D.ORNOT.MASK,D_ALU,R[0C]_ALU,SC_SC+K[.8]
U 1349. 0001.001C.0180.0AE8.0000.134A  ;3129  ALU_D.ORNOT.MASK,R[0D]_ALU
U 134A. 0018.0038.4180.0A90.0000.134B  ;3130  R[2]_K[.80]  ; SET UP LA AND LB VALUES.
U 134B. 0018.0038.4180.0A98.0000.10F0  ;3131  R[3]_K[.80]
U 10F0. 0000.003D.0180.0800.0000.1191  ;3132 =0 ERLOOP
U 10F1. 0000.003C.0180.0800.0000.10F2  ;3133  NOP
          ;3134 =0
U 10F2. 0000.003D.0180.0800.0000.119B  ;3135 ADDT1A: CALL,J/FPINIT
U 10F3. 0018.0038.2180.0800.0200.10F4  ;3136  VA_K[.14]  ; LOAD ADDD INTO IB
U 10F4. 0000.003D.0180.0800.0000.11A6  ;3137 =0 CALL,J/LOADIB
U 10F5. 0000.003C.0180.0800.0000.10F6  ;3138  NOP
U 10F6. 0000.003D.0180.0800.0000.11B0  ;3139 =0 CALL,J/ADDSIM  ; GO CALCULATE THE EXPECTED RESULT.
U 10F7. 0000.003C.0180.0800.0000.10F8  ;3140  NOP
U 10F8. 0000.003D.0180.0800.0000.11E4  ;3141 =0 CALL,J/UNSRES  ; PACK RESULT
U 10F9. 0000.003C.0180.0800.0000.10FA  ;3142  NOP
U 10FA. 0000.003D.0180.0800.0000.11EA  ;3143 =0 CALL,J/UNSAOP  ; PACK AR OPERAND
U 10FB. 0000.003C.0180.0800.0000.10FC  ;3144  NOP
U 10FC. 0000.003D.0180.0800.0000.11F0  ;3145 =0 CALL,J/UNSBOP  ; PACK BR OPERAND
U 10FD. 0000.003C.0180.0800.0000.10FE  ;3146  NOP
U 10FE. 0000.003D.0180.0800.0000.11F6  ;3147 =0 CALL,J/EXEC  ; GO EXECUTE THE FALU A+B
          ;3148      ; AND GET THE RESULTS.
U 10FF. 0810.0038.0180.0960.0000.134C  ;3149  D_RC[0C]  ; CHECK THE RESULTS
U 134C. 0010.0038.01C0.0970.0000.134D  ;3150  Q_RC[0E]
U 134D. 001D.0000.0180.0800.0010.134E  ;3151  ALU_D-Q,CLK.UBCC
U 134E. 0000.013C.0180.0800.0000.110A  ;3152  Z?
U 110A. 0000.003D.0180.0800.0000.1192  ;3153 =0 ERROR1   ; FALU HI RESULT INCORRECT
U 110B. 0810.0038.0180.0958.0000.134F  ;3154  D_RC[0B]
U 134F. 0010.0038.01C0.0968.0000.1350  ;3155  Q_RC[0D]
U 1350. 001D.0000.0180.0800.0010.1351  ;3156  ALU_D-Q,CLK.UBCC
U 1351. 0000.013C.0180.0800.0000.1110  ;3157  Z?
U 1110. 0000.003D.0180.0800.0000.1192  ;3158 =0 ERROR1   ; FALU LO RESULT INCORRECT
U 1111. 0000.003C.0180.0800.0000.1112  ;3159  NOP
U 1112. 0000.003D.0180.0800.0000.11C1  ;3160 =0 CALL,J/INCB  ; INCREMENT BR OPERAND
U 1113. 0000.003C.0180.0A08.0010.1352  ;3161  ALU_R[1],CLK.UBCC ; TRIED ALL BR OPERANDS?
U 1352. 0000.013C.0180.0800.0000.1114  ;3162  Z?
U 1114. 0018.0000.0580.0A88.0000.10F2  ;3163 =0 R[1]_LA-K[.1],J/ADDT1A ; IF NOT CONTINUE
U 1115. 0018.0038.5580.0A88.0000.1353  ;3164  R[1]_K[.3F]  ; REINITIALIZE BR OPERAND
U 1353. 0800.003C.0180.0A70.0000.1354  ;3165  D_R[0E]
U 1354. 0001.003C.0180.09B0.0000.1355  ;3166  RC[6]_D
U 1355. 0003.003C.0180.09A8.0000.1116  ;3167  RC[5]_0
U 1116. 0000.003D.0180.0800.0000.11BB  ;3168 =0 CALL,J/INCA  ; INCREMENT AR OPERAND
U 1117. 0000.003C.0180.0A00.0010.1356  ;3169  ALU_R[0],CLK.UBCC ; TRIED ALL AR OPERANDS?
U 1356. 0000.013C.0180.0800.0000.1118  ;3170  Z?
U 1118. 0018.0000.0580.0A80.0000.10F2  ;3171 =0 R[0]_LA-K[.1],J/ADDT1A ; IF NOT CONTINUE
U 1119. 0000.003C.0180.0800.0000.111A  ;3172  NOP
          ;3173 ;
```

```
            ;3174  .PAGE "TEST 273 FALU A+B TEST 2"
            ;3175 ;*****************************************************************
            ;3176 ;++
            ;3177 ; FALU A+B TEST 2
            ;3178 ;
            ;3179 ; TEST DESCRIPTION
            ;3180 ; THIS TEST PERFORMS FALU ADD'S WITH OPERANDS WHICH
            ;3181 ; TRY EVERY COMBINATION OF INPUT PATTERNS TO THE 5 BIT
            ;3182 ; FALU SLICES:
            ;3183 ;  FALU <61:57>
            ;3184 ;  FALU <55:51>
            ;3185 ;  FALU <47:43>
            ;3186 ;  FALU <39:35>
            ;3187 ;  FALU <31:27>
            ;3188 ;  FALU <23:19>
            ;3189 ;  FALU <15:11>
            ;3190 ;  FALU <07:03>
            ;3191 ;
            ;3192 ; LOGIC DESCRIPTION
            ;3193 ;
            ;3194 ; FPA UCODE USED
            ;3195 ; LOCATION CONTENTS
            ;3196 ; 4F:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/39
            ;3197 ; 39:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/3A
            ;3198 ; 3A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/3B
            ;3199 ; 3B:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/3C
            ;3200 ; 3C:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/3D
            ;3201 ; 3D:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/3E
            ;3202 ; 3E:  FADC/A.B,BSC/FAL.HL,NAD/3F
            ;3203 ; 3F:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
            ;3204 ;
            ;3205 ; ERROR DESCRIPTION
            ;3206 ; EXPECTED RESULT HI, FROM BUS A
            ;3207 ; EXPECTED RESULT LO, FROM BUS A
            ;3208 ; GOT RESULT HI, FROM BUS A
            ;3209 ; GOT RESULT LO, FROM BUS A
            ;3210 ; EXPECTED OUTPUT OF FALU HI
            ;3211 ; EXPECTED OUTPUT OF FALU LO
            ;3212 ; CONTENTS OF AR HI
            ;3213 ; CONTENTS OF AR LO
            ;3214 ; CONTENTS OF BR HI
            ;3215 ; CONTENTS OF BR LO
            ;3216 ;
            ;3217 ; SYNC POINT DESCRIPTION
            ;3218 ;
            ;3219 ;--
            ;3220 ;*****************************************************************
            ;3221 =0
U 111A, 0000.003D.0180.0800.0000.117F  ;3222 ADDT2: NEWTST
U 111B, 0018.0038.F580.09F8.0000.1357  ;3223  RC[0F]_K[.A]
U 1357, 0018.0038.5580.0A80.0000.1358  ;3224  R[0]_K[.3F]  ; INITIALIZE SOME COUNTERS.
U 1358, 0018.0038.5580.0A88.0000.1359  ;3225  R[1]_K[.3F]
U 1359, 0003.003C.01F8.0AF8.0000.135A  ;3226  Q_0,R[0F]_0  ; INITIALIZE AR OPERAND
U 135A, 0003.003C.03B0.09B8.0000.135B  ;3227  Q_Q.RIGHT,SI/7,RC[7]_0 ; AND BR OPERAND
U 135B, 0003.003C.01B0.09A8.0000.135C  ;3228  Q_Q.RIGHT,SI/ZERO,RC[5]_0
```

```
U 135C. 0001,203C,0180,09C0,0000,135D  ;3229  RC[8]_Q
U 135D. 0001,203C,0180,09B0,0000,135E  ;3230  RC[6]_Q
U 135E. 0001,203C,0180,0AF0,0000,135F  ;3231  R[0E]_Q
U 135F. 0000,003C,0D80,0800,0084,7360  ;3232  SC_K[.3]  ; SET UP INCREMENT TO OPERANDS.
U 1360. 0803,001C,0180,0800,0084,9361  ;3233  ALU_NOT.MASK,D_ALU,SC_SC+K[.8]
U 1361. 0801,001C,0180,0800,0084,9362  ;3234  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 1362. 0801,001C,0180,0800,0084,9363  ;3235  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 1363. 0001,001C,B980,0AE8,0084,7364  ;3236  ALU_D.ORNOT.MASK,R[0D]_ALU,SC_K[.19]
U 1364. 0001,001C,0180,0AE0,0000,1365  ;3237  ALU_D.ORNOT.MASK,R[0C]_ALU
U 1365. 0018,0038,4180,0A90,0000,1366  ;3238  R[2]_K[.80]  ; SET UP LA AND LB VALUES.
U 1366. 0018,0038,4180,0A98,0000,111C  ;3239  R[3]_K[.80]
U 111C. 0000,003D,0180,0800,0000,1191  ;3240 =0 ERLOOP
U 111D. 0000,003C,0180,0800,0000,111E  ;3241  NOP
        ;3242 =0
U 111E. 0000,003D,0180,0800,0000,119B  ;3243 ADDT2A: CALL,J/FPINIT
U 111F. 0018,0038,2180,0800,0200,1120  ;3244  VA_K[.14]  ; LOAD ADDD INTO IB
U 1120. 0000,003D,0180,0800,0000,11A6  ;3245 =0 CALL,J/LOADIB
U 1121. 0000,003C,0180,0800,0000,1122  ;3246  NOP
U 1122. 0000,003D,0180,0800,0000,11B0  ;3247 =0 CALL,J/ADDSIM  ; GO CALCULATE THE EXPECTED RESULT.
U 1123. 0000,003C,0180,0800,0000,1124  ;3248  NOP
U 1124. 0000,003D,0180,0800,0000,11E4  ;3249 =0 CALL,J/UNSRES  ; PACK RESULT
U 1125. 0000,003C,0180,0800,0000,1126  ;3250  NOP
U 1126. 0000,003D,0180,0800,0000,11EA  ;3251 =0 CALL,J/UNSAOP  ; PACK AR OPERAND
U 1127. 0000,003C,0180,0800,0000,1128  ;3252  NOP
U 1128. 0000,003D,0180,0800,0000,11F0  ;3253 =0 CALL,J/UNSBOP  ; PACK BR OPERAND
U 1129. 0000,003C,0180,0800,0000,112A  ;3254  NOP
U 112A. 0000,003D,0180,0800,0000,11F6  ;3255 =0 CALL,J/EXEC  ; GO EXECUTE THE FALU A+B
        ;3256     ; AND GET THE RESULTS.
U 112B. 0810,0038,0180,0960,0000,1367  ;3257  D_RC[0C]  ; CHECK THE RESULTS
U 1367. 0010,0038,01C0,0970,0000,1368  ;3258  Q_RC[0E]
U 1368. 001D,0000,0180,0800,0010,1369  ;3259  ALU_D-Q,CLK.UBCC
U 1369. 0000,013C,0180,0800,0000,112C  ;3260  Z?
U 112C. 0000,003D,0180,0800,0000,1192  ;3261 =0 ERROR1   ; FALU HI RESULT INCORRECT
U 112D. 0810,0038,0180,0958,0000,136A  ;3262  D_RC[0B]
U 136A. 0010,0038,01C0,0968,0000,136B  ;3263  Q_RC[0D]
U 136B. 001D,0000,0180,0800,0010,136C  ;3264  ALU_D-Q,CLK.UBCC
U 136C. 0000,013C,0180,0800,0000,112E  ;3265  Z?
U 112E. 0000,003D,0180,0800,0000,1192  ;3266 =0 ERROR1   ; FALU LO RESULT INCORRECT
U 112F. 0000,003C,0180,0800,0000,1130  ;3267  NOP
U 1130. 0000,003D,0180,0800,0000,11C1  ;3268 =0 CALL,J/INCB  ; INCREMENT BR OPERAND
U 1131. 0000,003C,0180,0A08,0010,136D  ;3269  ALU_R[1],CLK.UBCC ; TRIED ALL BR OPERANDS?
U 136D. 0000,013C,0180,0800,0000,1132  ;3270  Z?
U 1132. 0018,0000,0580,0A88,0000,111E  ;3271 =0 R[1]_LA-K[.1],J/ADDT2A ; IF NOT CONTINUE
U 1133. 0018,0038,5580,0A88,0000,136E  ;3272  R[1]_K[.3F]  ; REINITIALIZE BR OPERAND
U 136E. 0800,003C,0180,0A70,0000,136F  ;3273  D_R[0E]
U 136F. 0001,003C,0180,09B0,0000,1370  ;3274  RC[6]_D
U 1370. 0003,003C,0180,09A8,0000,1134  ;3275  RC[5]_0
U 1134. 0000,003D,0180,0800,0000,11BB  ;3276 =0 CALL,J/INCA  ; INCREMENT AR OPERAND
U 1135. 0000,003C,0180,0A00,0010,1371  ;3277  ALU_R[0],CLK.UBCC ; TRIED ALL AR OPERANDS?
U 1371. 0000,013C,0180,0800,0000,1136  ;3278  Z?
U 1136. 0018,0000,0580,0A80,0000,111E  ;3279 =0 R[0]_LA-K[.1],J/ADDT2A ; IF NOT CONTINUE
U 1137. 0000,003C,0180,0800,0000,1138  ;3280  NOP
```

```
;3281  .PAGE "TEST 274 FALU A-B TEST 1"
;3282 ;*************************************************************
;3283 ;++
;3284 ; FALU A-B TEST 1
;3285 ;
;3286 ; TEST DESCRIPTION
;3287 ; THIS TEST PERFORMS FALU SUB'S WITH OPERANDS WHICH
;3288 ; TRY EVERY COMBINATION OF INPUT PATTERNS TO THE 5 BIT
;3289 ; FALU SLICES:
;3290 ;  FALU <59:55>
;3291 ;  FALU <51:47>
;3292 ;  FALU <43:39>
;3293 ;  FALU <35:31>
;3294 ;  FALU <27:23>
;3295 ;  FALU <19:15>
;3296 ;  FALU <11:07>
;3297 ;
;3298 ; LOGIC DESCRIPTION
;3299 ;
;3300 ; FPA UCODE USED
;3301 ; LOCATION CONTENTS
;3302 ; 4F:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/39
;3303 ; 39:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/3A
;3304 ; 3A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/3B
;3305 ; 3B:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/3C
;3306 ; 3C:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/3D
;3307 ; 3D:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/3E
;3308 ; 3E:  FADC/A.B,BSC/FAL.HL,NAD/3F
;3309 ; 3F:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
;3310 ;
;3311 ; ERROR DESCRIPTION
;3312 ; EXPECTED RESULT HI, FROM BUS A
;3313 ; EXPECTED RESULT LO, FROM BUS A
;3314 ; GOT RESULT HI, FROM BUS A
;3315 ; GOT RESULT LO, FROM BUS A
;3316 ; EXPECTED OUTPUT OF FALU HI
;3317 ; EXPECTED OUTPUT OF FALU LO
;3318 ; CONTENTS OF AR HI
;3319 ; CONTENTS OF AR LO
;3320 ; CONTENTS OF BR HI
;3321 ; CONTENTS OF BR LO
;3322 ;
;3323 ; SYNC POINT DESCRIPTION
;3324 ;
;3325 ;--
;3326 ;*************************************************************
;3327 =0
U 1138. 0000.003D.0180.0800.0000.117F  ;3328 SUBT1: NEWTST
U 1139. 0018.0038.F580.09F8.0000.1372  ;3329  RC[0F]_K[.A]
U 1372. 0018.0038.5580.0A80.0000.1373  ;3330  R[0]_K[.3F]  ; INITIALIZE SOME COUNTERS.
U 1373. 0018.0038.5580.0A88.0000.1374  ;3331  R[1]_K[.3F]
U 1374. 0003.003C.01F8.0AF8.0000.1375  ;3332  Q_0,R[0F]_0  ; INITIALIZE AR OPERAND
U 1375. 0000.003C.03B0.0800.0000.1376  ;3333  Q_Q.RIGHT,SI/7  ; AND BR OPERAND
U 1376. 0003.003C.01B0.09B8.0000.1377  ;3334  Q_Q.RIGHT,SI/ZERO,RC[7]_0
U 1377. 0001.203C.0180.09C0.0000.1378  ;3335  RC[8]_Q
```

```
U 1378. 0001,203C,0180,0AF0,0000,1379  ;3336  R[0E]_Q
U 1379. 0818,0038,5DF8,0800,0000,137A  ;3337  D_K[.7],Q_0
U 137A. 0000,003C,1180,0800,0084,737B  ;3338  SC_K[.4]
U 137B. 0D00,003C,0180,0800,0000,137C  ;3339  D_DAL.SC
U 137C. 0F00,003C,71E0,0800,0084,737D  ;3340  Q_D,D_0,SC_K[.FFF8]
U 137D. 0D00,003C,0180,0800,0000,137E  ;3341  D_DAL.SC
U 137E. 0D00,003C,0180,0800,0000,137F  ;3342  D_DAL.SC
U 137F. 0D00,003C,0180,0800,0000,1380  ;3343  D_DAL.SC
U 1380. 0001,003C,0180,09A8,0000,1381  ;3344  RC[5]_D
U 1381. 0D00,003C,0180,0800,0000,1382  ;3345  D_DAL.SC
U 1382. 0001,003C,0180,09B0,0000,1383  ;3346  RC[6]_D
U 1383. 0000,003C,5D80,0800,0084,7384  ;3347  SC_K[.7]  ; SET UP INCREMENT TO OPERANDS.
U 1384. 0803,001C,0180,0800,0084,9385  ;3348  ALU_NOT.MASK,D_ALU,SC_SC+K[.8]
U 1385. 0801,001C,0180,0800,0084,9386  ;3349  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 1386. 0801,001C,0180,0AE0,0084,9387  ;3350  ALU_D.ORNOT.MASK,D_ALU,R[0C]_ALU,SC_SC+K[.8]
U 1387. 0001,001C,0180,0AE8,0000,1388  ;3351  ALU_D.ORNOT.MASK,R[0D]_ALU
U 1388. 0818,0038,4180,0A90,0000,1389  ;3352  R[2]_K[.80],D_K[.80] ; SET UP LA AND LB VALUES.
U 1389. 0019,0030,4580,0A98,0000,113A  ;3353  ALU_D[OR]K[.8000],R[3]_ALU
U 113A. 0000,003D,0180,0800,0000,1191  ;3354 =0 ERLOOP
U 113B. 0000,003C,0180,0800,0000,113C  ;3355  NOP
        ;3356 =0
U 113C. 0000,003D,0180,0800,0000,119B  ;3357 SUBT1A: CALL,J/FPINIT
U 113D. 0018,0038,2180,0800,0200,113E  ;3358  VA_K[.14]  ; LOAD ADDD INTO IB
U 113E. 0000,003D,0180,0800,0000,11A6  ;3359 =0 CALL,J/LOADIB
U 113F. 0000,003C,0180,0800,0000,1140  ;3360  NOP
U 1140. 0000,003D,0180,0800,0000,11B6  ;3361 =0 CALL,J/SUBSIM  ; GO CALCULATE THE EXPECTED RESULT.
U 1141. 0000,003C,0180,0800,0000,1142  ;3362  NOP
U 1142. 0000,003D,0180,0800,0000,11E4  ;3363 =0 CALL,J/UNSRES  ; PACK RESULT
U 1143. 0000,003C,0180,0800,0000,1144  ;3364  NOP
U 1144. 0000,003D,0180,0800,0000,11EA  ;3365 =0 CALL,J/UNSAOP  ; PACK AR OPERAND
U 1145. 0000,003C,0180,0800,0000,1146  ;3366  NOP
U 1146. 0000,003D,0180,0800,0000,11F0  ;3367 =0 CALL,J/UNSBOP  ; PACK BR OPERAND
U 1147. 0000,003C,0180,0800,0000,1148  ;3368  NOP
U 1148. 0000,003D,0180,0800,0000,11F6  ;3369 =0 CALL,J/EXEC  ; GO EXECUTE THE FALU A+B
        ;3370      ; AND GET THE RESULTS.
U 1149. 0810,0038,0180,0960,0000,138A  ;3371  D_RC[0C]  ; CHECK THE RESULTS
U 138A. 0010,0038,01C0,0970,0000,138B  ;3372  Q_RC[0E]
U 138B. 001D,0000,0180,0800,0010,138C  ;3373  ALU_D-Q,CLK.UBCC
U 138C. 0000,013C,0180,0800,0000,114A  ;3374  Z?
U 114A. 0000,003D,0180,0800,0000,1192  ;3375 =0 ERROR1   ; FALU HI RESULT INCORRECT
U 114B. 0810,0038,0180,0958,0000,138D  ;3376  D_RC[0B]
U 138D. 0010,0038,01C0,0968,0000,138E  ;3377  Q_RC[0D]
U 138E. 001D,0000,0180,0800,0010,138F  ;3378  ALU_D-Q,CLK.UBCC
U 138F. 0000,013C,0180,0800,0000,114C  ;3379  Z?
U 114C. 0000,003D,0180,0800,0000,1192  ;3380 =0 ERROR1   ; FALU LO RESULT INCORRECT
U 114D. 0000,003C,0180,0800,0000,114E  ;3381  NOP
U 114E. 0000,003D,0180,0800,0000,11C1  ;3382 =0 CALL,J/INCB  ; INCREMENT BR OPERAND
U 114F. 0000,003C,0180,0A08,0010,1390  ;3383  ALU_R[1],CLK.UBCC ; TRIED ALL BR OPERANDS?
U 1390. 0000,013C,0180,0800,0000,1150  ;3384  Z?
U 1150. 0018,0000,0580,0A88,0000,113C  ;3385 =0 R[1]_LA-K[.1],J/SUBT1A ; IF NOT CONTINUE
U 1151. 0018,0038,5580,0A88,0000,1391  ;3386  R[1]_K[.3F]  ; REINITIALIZE BR OPERAND
U 1391. 0800,003C,0180,0A70,0000,1392  ;3387  D_R[0E]
U 1392. 0001,003C,0180,09B0,0000,1393  ;3388  RC[6]_D
U 1393. 0003,003C,0180,09A8,0000,1152  ;3389  RC[5]_0
U 1152. 0000,003D,0180,0800,0000,11BB  ;3390 =0 CALL,J/INCA  ; INCREMENT AR OPERAND
```

```
ZZ-ESKAR-V22  TEST 274 FALU A-B TEST 1                                                    SEQ 2580
ESKARSF.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:52                     Page   8

U 1153, 0000,003C,0180,0A00,0010,1394  ;3391  ALU_R[0],CLK.UBCC ; TRIED ALL AR OPERANDS?
U 1394, 0000,013C,0180,0800,0000,1156  ;3392  Z?
U 1156, 0018,0000,0580,0A80,0000,113C  ;3393 =0 R[0]_LA-K[.1],J/SUBT1A ; IF NOT CONTINUE
U 1157, 0000,003C,0180,0800,0000,1158  ;3394  NOP
```

```
;3395  .PAGE "TEST 275 FALU A-B TEST 2"
;3396 ;*************************************************************
;3397 ;**
;3398 ; FALU A-B TEST 2
;3399 ;
;3400 ; TEST DESCRIPTION
;3401 ; THIS TEST PERFORMS FALU SUB'S WITH OPERANDS WHICH
;3402 ; TRY EVERY COMBINATION OF INPUT PATTERNS TO THE 5 BIT
;3403 ; FALU SLICES:
;3404 ;  FALU <61:57>
;3405 ;  FALU <55:51>
;3406 ;  FALU <47:43>
;3407 ;  FALU <39:35>
;3408 ;  FALU <31:27>
;3409 ;  FALU <23:19>
;3410 ;  FALU <15:11>
;3411 ;  FALU <07:03>
;3412 ;
;3413 ; LOGIC DESCRIPTION
;3414 ;
;3415 ; FPA UCODE USED
;3416 ; LOCATION CONTENTS
;3417 ; 4F:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/39
;3418 ; 39:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/3A
;3419 ; 3A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/3B
;3420 ; 3B:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/3C
;3421 ; 3C:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/3D
;3422 ; 3D:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/3E
;3423 ; 3E:  FADC/A.B,BSC/FAL.HL,NAD/3F
;3424 ; 3F:  FADC/A.B,BSC/FAL.LH,NAD/PSTALL
;3425 ;
;3426 ; ERROR DESCRIPTION
;3427 ; EXPECTED RESULT HI, FROM BUS A
;3428 ; EXPECTED RESULT LO, FROM BUS A
;3429 ; GOT RESULT HI, FROM BUS A
;3430 ; GOT RESULT LO, FROM BUS A
;3431 ; EXPECTED OUTPUT OF FALU HI
;3432 ; EXPECTED OUTPUT OF FALU LO
;3433 ; CONTENTS OF AR HI
;3434 ; CONTENTS OF AR LO
;3435 ; CONTENTS OF BR HI
;3436 ; CONTENTS OF BR LO
;3437 ;
;3438 ; SYNC POINT DESCRIPTION
;3439 ;
;3440 ;--
;3441 ;*************************************************************
;3442 =0
U 1158, 0000,003D,0180,0800,0000,117F  ;3443 SUBT2: NEWTST
U 1159, 0018,0038,F580,09F8,0000,1395  ;3444  RC[0F]_K[.A]
U 1395, 0018,0038,5580,0A80,0000,1396  ;3445  R[0]_K[.3F]  ; INITIALIZE SOME COUNTERS.
U 1396, 0018,0038,5580,0A88,0000,1397  ;3446  R[1]_K[.3F]
U 1397, 0003,003C,01F8,0AF8,0000,1398  ;3447  Q_0,R[0F]_0  ; INITIALIZE AR OPERAND
U 1398, 0000,003C,03B0,0800,0000,1399  ;3448  Q_Q.RIGHT,SI/7  ; AND BR OPERAND
U 1399, 0003,003C,01B0,09B8,0000,139A  ;3449  Q_Q.RIGHT,SI/ZERO,RC[7]_0
```

```
U 139A, 0001,203C,0180,09C0,0000,139B  ;3450  RC[8]_Q
U 139B, 0001,203C,0180,0AF0,0000,139C  ;3451  R[0E]_Q
U 139C, 0F18,0038,5DC0,0800,0000,139D  ;3452  D_0,Q_K[.7]
U 139D, 0000,003C,7180,0800,0084,739E  ;3453  SC_K[.FFF8]
U 139E, 0D00,003C,0180,0800,0000,139F  ;3454  D_DAL.SC
U 139F, 0D00,003C,0180,0800,0000,13A0  ;3455  D_DAL.SC
U 13A0, 0D00,003C,0180,0800,0000,13A1  ;3456  D_DAL.SC
U 13A1, 0001,003C,0180,09A8,0000,13A2  ;3457  RC[5]_D
U 13A2, 0018,0038,31C0,0800,0000,13A3  ;3458  Q_K[.40]
U 13A3, 0019,2014,05C0,0800,0000,13A4  ;3459  ALU_Q+K[.1],Q_ALU
U 13A4, 0D00,003C,0180,0800,0000,13A5  ;3460  D_DAL.SC
U 13A5, 0001,003C,0180,09B0,0000,13A6  ;3461  RC[6]_D
U 13A6, 0000,003C,0D80,0800,0084,73A7  ;3462  SC_K[.3]  ; SET UP INCREMENT TO OPERANDS.
U 13A7, 0803,001C,0180,0800,0084,93A8  ;3463  ALU_NOT.MASK,D_ALU,SC_SC+K[.8]
U 13A8, 0801,001C,0180,0800,0084,93A9  ;3464  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 13A9, 0801,001C,0180,0800,0084,93AA  ;3465  ALU_D.ORNOT.MASK,D_ALU,SC_SC+K[.8]
U 13AA, 0001,001C,B980,0AE8,0084,73AB  ;3466  ALU_D.ORNOT.MASK,R[0D]_ALU,SC_K[.19]
U 13AB, 0001,001C,0180,0AE0,0000,13AC  ;3467  ALU_D.ORNOT.MASK,R[0C]_ALU
U 13AC, 0818,0038,4180,0A90,0000,13AD  ;3468  R[2]_K[.80],D_ALU ; SET UP LA AND LB VALUES.
U 13AD, 0019,0030,4580,0A98,0000,115A  ;3469  ALU_D[OR]K[.8000],R[3]_ALU
U 115A, 0000,003D,0180,0800,0000,1191  ;3470 =0 ERLOOP
U 115B, 0000,003C,0180,0800,0000,115C  ;3471  NOP
          ;3472 =0
U 115C, 0000,003D,0180,0800,0000,119B  ;3473 SUBT2A: CALL,J/FPINIT
U 115D, 0018,0038,2180,0800,0200,115E  ;3474  VA_K[.14]  ; LOAD ADDD INTO IB
U 115E, 0000,003D,0180,0800,0000,11A6  ;3475 =0 CALL,J/LOADIB
U 115F, 0000,003C,0180,0800,0000,1160  ;3476  NOP
U 1160, 0000,003D,0180,0800,0000,11B6  ;3477 =0 CALL,J/SUBSIM  ; GO CALCULATE THE EXPECTED RESULT.
U 1161, 0000,003C,0180,0800,0000,1162  ;3478  NOP
U 1162, 0000,003D,0180,0800,0000,11E4  ;3479 =0 CALL,J/UNSRES  ; PACK RESULT
U 1163, 0000,003C,0180,0800,0000,1164  ;3480  NOP
U 1164, 0000,003D,0180,0800,0000,11EA  ;3481 =0 CALL,J/UNSAOP  ; PACK AR OPERAND
U 1165, 0000,003C,0180,0800,0000,1166  ;3482  NOP
U 1166, 0000,003D,0180,0800,0000,11F0  ;3483 =0 CALL,J/UNSBOP  ; PACK BR OPERAND
U 1167, 0000,003C,0180,0800,0000,1168  ;3484  NOP
U 1168, 0000,003D,0180,0800,0000,11F6  ;3485 =0 CALL,J/EXEC  ; GO EXECUTE THE FALU A+B
          ;3486     ; AND GET THE RESULTS.
U 1169, 0810,0038,0180,0960,0000,13AE  ;3487  D_RC[0C]  ; CHECK THE RESULTS
U 13AE, 0010,0038,01C0,0970,0000,13AF  ;3488  Q_RC[0E]
U 13AF, 001D,0000,0180,0800,0010,13B0  ;3489  ALU_D-Q,CLK.UBCC
U 13B0, 0000,013C,0180,0800,0000,116A  ;3490  Z?
U 116A, 0000,003D,0180,0800,0000,1192  ;3491 =0 ERROR1   ; FALU HI RESULT INCORRECT
U 116B, 0810,0038,0180,0958,0000,13B1  ;3492  D_RC[0B]
U 13B1, 0010,0038,01C0,0968,0000,13B2  ;3493  Q_RC[0D]
U 13B2, 001D,0000,0180,0800,0010,13B3  ;3494  ALU_D-Q,CLK.UBCC
U 13B3, 0000,013C,0180,0800,0000,116C  ;3495  Z?
U 116C, 0000,003D,0180,0800,0000,1192  ;3496 =0 ERROR1   ; FALU LO RESULT INCORRECT
U 116D, 0000,003C,0180,0800,0000,116E  ;3497  NOP
U 116E, 0000,003D,0180,0800,0000,11C1  ;3498 =0 CALL,J/INCB  ; INCREMENT BR OPERAND
U 116F, 0000,003C,0180,0A08,0010,13B4  ;3499  ALU_R[1],CLK.UBCC ; TRIED ALL BR OPERANDS?
U 13B4, 0000,013C,0180,0800,0000,1170  ;3500  Z?
U 1170, 0018,0000,0580,0A88,0000,115C  ;3501 =0 R[1]_LA-K[.1],J/SUBT2A ; IF NOT CONTINUE
U 1171, 0018,0038,5580,0A88,0000,13B5  ;3502  R[1]_K[.3F]  ; REINITIALIZE BR OPERAND
U 13B5, 0800,003C,0180,0A70,0000,13B6  ;3503  D_R[0E]
U 13B6, 0001,003C,0180,09B0,0000,13B7  ;3504  RC[6]_D
```

```
U 13B7, 0003.003C.0180.09A8.0000.1172  ;3505  RC[5]_0
U 1172, 0000.003D.0180.0800.0000.11BB  ;3506 =0 CALL,J/INCA  ; INCREMENT AR OPERAND
U 1173, 0000.003C.0180.0A00.0010.13B8  ;3507  ALU_R[0],CLK.UBCC ; TRIED ALL AR OPERANDS?
U 13B8, 0000.013C.0180.0800.0000.1174  ;3508  Z?
U 1174, 0018.0000.0580.0A80.0000.115C  ;3509 =0 R[0]_LA-K[.1],J/SUBT2A ; IF NOT CONTINUE
U 1175, 0000.003C.0180.0800.0000.1176  ;3510  NOP
          ;3511
```

```
            ;3512  .PAGE "TEST 276 RESERVED"
            ;3513 =0
U 1176. 0000.003D.0180.0800.0000.117F  ;3514 D1: NEWTST
U 1177. 0000.003C.0180.0800.0000.1178  ;3515  NOP
            ;3516
```

```
            ;3517  .PAGE "DUMMY NEWTST"
            ;3518 =0
U 1178. 0000.003D.0180.0800.0000.117F  ;3519 D3: NEWTST
```

ZZ-ESKAR-V22 Line Number Index
ESKAR5F.MCR MICRO2 1P(00) 26-JUL-85 17:03:52
: Location / Line Number Index


```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 1918= 1919=
U 1008  1995= 1996= 1998= 1999= 2000= 2001= 2058= 2059=
U 1010  2003= 2004= 2023= 2024= 2172= 2173= 2186= 2187=
U 1018  2200= 2201= 2211= 2212= 2260= 2261= 2077= 2078=
U 1020  2268= 2269= 2271= 2272= 2274= 2275= 2301= 2302=
U 1028  2304= 2305= 2308= 2309= 2344= 2345= 2095= 2096=
U 1030  2349= 2350= 2352= 2353= 2354= 2355= 2356= 2357=
U 1038  2359= 2360= 2379= 2380= 2395= 2396= 2117= 2118=
U 1040  2398= 2399= 2402= 2403= 2467= 2468= 1875 2288=
U 1048  2478= 2479= 2482= 2483= 2485= 2486= 1939: 2289=
U 1050  2503= 2504= 2505= 2506= 2507= 2508= 1876 2373=
U 1058  2509= 2510= 1942: 1877 2526= 2527= 1945: 2374=
U 1060  2540= 2541= 2542= 2543= 2544= 2545= 1878 2985=
U 1068  2546= 2547= 2563= 2564= 2577= 2578= 1879 2986=
U 1070  2579= 2580= 2581= 2582= 2583= 2584= 2600= 2601=
U 1078  2614= 2615= 2616= 2617= 2618= 2619= 2620= 2621=
U 1080  2637= 2638= 2651= 2652= 2653= 2654= 2655= 2656=
U 1088  2657= 2658= 2674= 2675= 2688= 2689= 2690= 2691=
U 1090  2692= 2693= 2694= 2695= 2711= 2712= 2725= 2726=
U 1098  2727= 2728= 2729= 2730= 2731= 2732= 2745= 2746=
U 10A0  2759= 2760= 2761= 2762= 2763= 2764= 2765= 2766=
U 10A8  2779= 2780= 2793= 2794= 2795= 2796= 2797= 2798=
U 10B0  2799= 2800= 2816= 2817= 2830= 2831= 2832= 2833=
U 10B8  2834= 2835= 2836= 2837= 2853= 2854= 2867= 2868=
U 10C0  2869= 2870= 2871= 2872= 2873= 2874= 2890= 2891=
U 10C8  2904= 2905= 2906= 2907= 2908= 2909= 2910= 2911=
U 10D0  2924= 2925= 2957= 2958= 2962= 2963= 2965= 2966=
U 10D8  2967= 2968= 2970= 2971= 2990= 2991= 3003= 3004=
U 10E0  3006= 3007= 3010= 3011= 3041= 3042= 3049= 3050=
U 10E8  3051= 3052= 3054= 3055= 3066= 3067= 3115= 3116=
U 10F0  3132= 3133= 3135= 3136= 3137= 3138= 3139= 3140=
U 10F8  3141= 3142= 3143= 3144= 3145= 3146= 3147= 3149=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1880 3153= 3154= 1870: 1871: 1872: 1873:
U 1110  3158= 3159= 3160= 3161= 3163= 3164= 3168= 3169=
U 1118  3171= 3172= 3222= 3223= 3240= 3241= 3243= 3244=
U 1120  3245= 3246= 3247= 3248= 3249= 3250= 3251= 3252=
U 1128  3253= 3254= 3255= 3257= 3261= 3262= 3266= 3267=
U 1130  3268= 3269= 3271= 3272= 3276= 3277= 3279= 3280=
U 1138  3328= 3329= 3354= 3355= 3357= 3358= 3359= 3360=
U 1140  3361= 3362= 3363= 3364= 3365= 3366= 3367= 3368=
U 1148  3369= 3371= 3375= 3376= 3380= 3381= 3382= 3383=
U 1150  3385= 3386= 3390= 3391= 1900 1965: 3393= 3394=
U 1158  3443= 3444= 3470= 3471= 3473= 3474= 3475= 3476=
U 1160  3477= 3478= 3479= 3480= 3481= 3482= 3483= 3484=
U 1168  3485= 3487= 3491= 3492= 3496= 3497= 3498= 3499=
U 1170  3501= 3502= 3506= 3507= 3509= 3510= 3514= 3515=
U 1178  3519= 1901 1902 1903 1904 1905 1906 1916
U 1180  1917 1920 1921 1922 1923 1924 1925 1926
U 1188  1927 1928 1929 1930 1931 1932 1933 1934
U 1190  1935 1936 1937 1938 1940 1941 1973 1974
U 1198  1980 1981 1982 1993 1994 1997 2002 2005
```

ZZ-ESKAR-V22 Line Number Index
ESKAR5F.MCR MICRO2 1P(00) 26-JUL-85 17:03:52 SEQ 2587 Page 9

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0 2015 2016 2017 2025 2026 2027 2033 2034
U 11A8 2035 2036 2037 2038 2039 2040 2041 2042
U 11B0 2054 2055 2056 2057 2061 2062 2073 2074
U 11B8 2075 2076 2079 2091 2092 2093 2094 2098
U 11C0 2099 2112 2113 2114 2115 2116 2120 2121
U 11C8 2129 2130 2131 2132 2133 2134 2135 2136
U 11D0 2137 2138 2139 2140 2141 2142 2149 2150
U 11D8 2151 2152 2153 2154 2155 2156 2157 2158
U 11E0 2159 2160 2161 2162 2169 2170 2171 2174
U 11E8 2175 2176 2183 2184 2185 2188 2189 2190
U 11F0 2197 2198 2199 2202 2203 2204 2210 2213
U 11F8 2214 2215 2216 2217 2218 2219 2220 2221
U 1200 2222 2223 2224 2225 2226 2227 2228 2229
U 1208 2262 2263 2264 2265 2266 2267 2273 2276
U 1210 2277 2278 2279 2280 2281 2282 2283 2284
U 1218 2285 2286 2287 2290 2291 2292 2293 2294
U 1220 2295 2296 2297 2298 2299 2300 2303 2306
U 1228 2307 2346 2347 2348 2358 2361 2362 2363
U 1230 2364 2365 2366 2367 2368 2369 2370 2371
U 1238 2372 2375 2376 2377 2378 2381 2382 2383
U 1240 2384 2385 2386 2387 2388 2389 2390 2391
U 1248 2392 2393 2394 2397 2400 2401 2469 2470
U 1250 2471 2472 2473 2474 2475 2476 2477 2480
U 1258 2481 2484 2487 2488 2489 2490 2511 2512
U 1260 2513 2514 2515 2516 2517 2518 2519 2520
U 1268 2521 2522 2523 2524 2525 2548 2549 2550
U 1270 2551 2552 2553 2554 2555 2556 2557 2558
U 1278 2559 2560 2561 2562 2585 2586 2587 2588
U 1280 2589 2590 2591 2592 2593 2594 2595 2596
U 1288 2597 2598 2599 2622 2623 2624 2625 2626
U 1290 2627 2628 2629 2630 2631 2632 2633 2634
U 1298 2635 2636 2659 2660 2661 2662 2663 2664
U 12A0 2665 2666 2667 2668 2669 2670 2671 2672
U 12A8 2673 2696 1966: 2697 2698 2699 2700 2701
U 12B0 2702 2703 2704 2705 2706 2707 2708 2709
U 12B8 2710 2733 2734 2735 2736 2737 2738 2739
U 12C0 2740 2741 2742 2743 2744 2767 2768 2769
U 12C8 2770 2771 2772 2773 2774 2775 2776 2777
U 12D0 2778 2801 2802 2803 2804 2805 2806 2807
U 12D8 2808 2809 2810 2811 2812 2813 2814 2815
U 12E0 2838 2839 2840 2841 2842 2843 2844 2845
U 12E8 2846 2847 2848 2849 2850 2851 2852 2875
U 12F0 2876 2877 2878 2879 2880 2881 2882 2883
U 12F8 2884 2885 2886 2887 2888 2889 2912 2913
U 1300 2914 2915 2916 2917 2918 2919 2920 2921
U 1308 2922 2923 2959 2960 2961 2969 2972 2973
U 1310 2974 2975 2976 2977 2978 2979 2980 2981
U 1318 2982 2983 2984 2987 2988 2989 2992 2993
U 1320 2994 2995 2996 2997 2998 2999 3000 3001
U 1328 3002 3005 3008 3009 3043 3044 3045 3046
U 1330 3047 3048 3053 3056 3057 3058 3059 3060
U 1338 3061 3062 3063 3064 3065 3117 3118 3119
U 1340 3120 3121 3122 3123 3124 3125 3126 3127

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 2588
ESKAR5F.MCR                    MICRO2  1P(00)    26-JUL-85  17:03:52          Page   9

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 1348  3128 3129 3130 3131 3150 3151 3152 3155
U 1350  3156 3157 3162 3165 3166 3167 3170 3224
U 1358  3225 3226 3227 3228 3229 3230 3231 3232
U 1360  3233 3234 3235 3236 3237 3238 3239 3258
U 1368  3259 3260 3263 3264 3265 3270 3273 3274
U 1370  3275 3278 3330 3331 3332 3333 3334 3335
U 1378  3336 3337 3338 3339 3340 3341 3342 3343
U 1380  3344 3345 3346 3347 3348 3349 3350 3351
U 1388  3352 3353 3372 3373 3374 3377 3378 3379
U 1390  3384 3387 3388 3389 3392 3445 3446 3447
U 1398  3448 3449 3450 3451 3452 3453 3454 3455
U 13A0  3456 3457 3458 3459 3460 3461 3462 3463
U 13A8  3464 3465 3466 3467 3468 3469 3488 3489
U 13B0  3490 3493 3494 3495 3500 3503 3504 3505
U 13B8  3508
U 13C0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR5F	969
Used	969
Remaining	

Total microwords used in memory U: 969
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR5F.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR60.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 2591

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2351 TEST 277 COMPLEMENT AR TEST
; 2498 TEST 278 COMPLEMENT BR TEST
; 2641 TEST 279 TEST FPA ERROR LATCH AND CPU BEN/6.
; 2974 TEST 27A RESERVED
; 2979 TEST 27B RESERVED
; 2984 TEST 27C RESERVED
; 2989 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR60
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7005  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7005  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7005  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7005  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7005  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7005  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7005  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7005  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7005  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7005  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7005  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7005  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7005  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.1002  ;1874 TRPH0: Q_ID[USTACK]
U 1002. 0C00.003C.01E0.0800.0000.1006  ;1875  Q_D,D_Q
U 1006. 0000.003C.8180.3C00.0000.100B  ;1876  ID[USTACK]_D
U 100B. 0C00.003C.01E0.0800.0000.1012  ;1877  Q_D,D_Q
U 1012. 0000.003C.01C0.0A78.0000.1016  ;1878  Q_R[0F]
U 1016. 0001.2024.0180.0800.0010.101B  ;1879  ALU_Q.ANDNOT.MASK.CLK.UBCC ; WAS TRAP EXPECTED?
U 101B. 0000.013C.0180.0800.0000.1004  ;1880  Z?    ; BRANCH IF NO
U 1004. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1005. 0018.0038.1D80.09E8.0000.100C   ;1897 TRPH1: RC[0D]_SC
U 100C. 0000.003D.D980.0809.0084.7113   ;1898 =0 SETRCF[.9]
U 100D. 0000.003C.49F0.2C00.0000.1022   ;1899  Q_ID[TBER0]
U 1022. 0001.203C.4DF0.2DB0.0000.1026   ;1900  RC[6]_Q,Q_ID[TBER1]
U 1026. 0001.203C.65F0.2DB8.0000.1032   ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 1032. 0001.203C.6DF0.2DD8.0000.1036   ;1902  RC[0B]_Q,Q_ID[FAULT]
U 1036. 0001.203C.75F0.2DE0.0000.103B   ;1903  RC[0C]_Q,Q_ID[MAINT]
U 103B. 0001.203C.79F0.2DC8.0000.1042   ;1904  RC[9]_Q,Q_ID[PARITY]
U 1042. 0001.203C.69F0.2DC0.0000.1046   ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 1046. 0001.203C.81F0.2DD0.0000.1000   ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.1109   ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 104A. 0000.003C.8580.0800.0084.705B   ;1916 SCOPE: SC_K[.C]
U 105B. 0803.001C.0180.0800.0000.1010   ;1917  ALU_NOT.MASK,D_ALU
U 1010. 0000.003D.7580.3C00.0000.1116   ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1011. 0000.003C.65FB.0800.0084.705F   ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 105F. 0818.0038.8D80.0800.0000.1063   ;1920  D_K[.1F]    ; ...
U 1063. 0D00.003C.0180.0800.0000.1067   ;1921  D_DAL.SC
U 1067. 0000.003C.3D80.3C00.0000.106B   ;1922  ID[PSL]_D    ; WRITE THE PSL
U 106B. 0818.0038.0580.0800.0000.10F3   ;1923  D_K[.1]     ; SETUP TO CLEAR THE CES REGISTER
U 10F3. 0D00.003C.0180.0800.0000.10F4   ;1924  D_DAL.SC
U 10F4. 0F00.003C.3180.3C00.0000.10F5   ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10F5. 0000.003C.0180.0800.0000.10F6   ;1926  NOP
U 10F6. 0000.003C.3980.3C00.0000.10F7   ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 10F7. 0000.003C.2980.3C00.0000.10F8   ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 10F8. 0000.003C.4980.3C00.0000.10F9   ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 10F9. 0818.0038.C180.0800.0000.10FA   ;1930  D_K[.FFFF]
U 10FA. 0000.003C.6580.3C00.0000.10FB   ;1931  ID[SBI.ERR]_D
U 10FB. 0F00.003C.6D80.3C00.0000.10FC   ;1932  ID[FAULT]_D,D_0
U 10FC. 0000.003C.6D80.3C00.0000.10FD   ;1933  ID[FAULT]_D
U 10FD. 0000.003C.6580.3C00.0000.10FE   ;1934  ID[SBI.ERR]_D
U 10FE. 0003.003C.0180.0AF8.0000.105E   ;1935  R[0F]_0,J/CALLCON   ; CALL THE CONSOLE
U 10FF. 0818.0038.0980.0800.0000.105E   ;1936 ERLOOP: D_K[.2],J/CALLCON   ; ERRLOOP CALL
U 1109. 0810.0038.0180.0978.0000.110A   ;1937 ERROR1: D_RC[0F]
U 110A. 0819.0034.4D80.0800.0000.104E   ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E   ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 110B. 0810.0038.0180.0978.0000.1110   ;1940 ERROR2: D_RC[0F]
U 1110. 0819.0034.4D80.0800.0000.105A   ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E   ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E   ;1945  ID[TXDB]_D,J/CALLCON   ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1   ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2   ;1950 13F1: NOP
```

```
U 13F2, 0000,003C,0180,0800,0000,13F3  ;1951 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;1952 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;1953 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;1954 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;1955 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;1956 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;1957 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;1958 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;1959 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;1960 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;1961 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;1962 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;1963 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;1964 13FF: NOP
U 1155, 0001,203E,59F0,2DE8,0000,0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001,203E,59F0,2DE8,0000,0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 1111, 0810,0038,4D80,0978,0000,1112  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1112, 0019,4016,4D80,09F8,0000,0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 1113, 0010,0038,01C0,0978,0000,1114  ;1980 SETRCF: Q_RC[0F]
U 1114, 0019,2034,4DC0,0800,0000,1115  ;1981  Q_Q.AND.K[.FF00]
U 1115, 0019,2032,1D80,09F8,0000,0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983 ;
            ;1984 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;1985 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;1986 ;   28: NAD/28   ; NOP
            ;1987 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;1988 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;1989 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;1990 ;  INITIALIZE ITSELF.
            ;1991 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;1992 ;
U 1116, 0818,0038,4580,0800,0000,1117  ;1993 FPINIT: D_K[.8000]
U 1117, 0000,003C,5D80,3C00,0000,1014  ;1994  ID[ACC.CS]_D   ; TURN ON FPA
U 1014, 0818,0039,2D80,0800,0000,111B  ;1995 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1015, 0000,003C,5980,3C00,0000,1118  ;1996  ID[ACC.2]_D
U 1118, 0000,00FC,0380,0800,0000,101C  ;1997  TRAP.FPA   ; TRAP FPA TO 28.
U 101C, 0018,0039,1980,0800,0200,1131  ;1998 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 101D, 0F00,003C,0180,0800,0000,1020  ;1999  D_0
U 1020, 0000,003D,0180,0800,0000,111B  ;2000 =0 CALL,J/GENTRA
U 1021, 0000,003C,5980,3C00,0000,1119  ;2001  ID[ACC.2]_D
U 1119, 0000,00FC,0380,0800,0000,1024  ;2002  TRAP.FPA   ; TRAP FPA TO 0
U 1024, 0818,0039,2D80,0800,0000,111B  ;2003 =0 D_K[.28],CALL,J/GENTRA
U 1025, 0000,003C,5980,3C00,0000,111A  ;2004  ID[ACC.2]_D
U 111A, 0000,00FE,0380,0800,0000,0001  ;2005  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
```

```
            ;2006 ;
            ;2007 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2008 ;x0
            ;2009 ;$ 0000.0000, 0000.0000
            ;2010 ;
            ;2011 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2012 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2013 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2014 ;
U 111B. 0000.003C.65F8.0800.0084.711C  ;2015 GENTRA: SC_K[.10],Q_0
U 111C. 0D00.003C.8D80.0800.0084.711D  ;2016  D_DAL.SC,SC_K[.1F]
U 111D. 0801.001E.0180.0800.0000.0001  ;2017  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2018 ;
            ;2019 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2020 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2021 ;
            ;2022 =0
U 102C. 0F00.003D.0180.0800.0000.111B  ;2023 FPIRD: D_0,CALL,J/GENTRA
U 102D. 0000.003C.5980.3C00.0000.111E  ;2024  ID[ACC.2]_D
U 111E. 0000.00FC.0380.0800.0000.111F  ;2025  TRAP.FPA  ; TRAP TO FPA 0
U 111F. 0000.003C.0180.0800.0000.1130  ;2026  NOP
U 1130. 0000.003E.0180.0800.0000.0001  ;2027  RETURN1
            ;2028 ;
            ;2029 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2030 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2031 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2032 ;
U 1131. 2000.003C.0180.0800.0000.1132  ;2033 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 1132. 3000.003C.0180.6000.0000.1133  ;2034  MCT/READ.V.NEWPC.IBC/START ; LOAD IPA AND START.
U 1133. 0000.003C.0180.F800.0000.1134  ;2035 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1134. 0000.003C.0180.F800.0000.1135  ;2036  MCT/ALLOW.IB.READ
U 1135. 0000.003C.0180.F800.0000.1136  ;2037  MCT/ALLOW.IB.READ
U 1136. 0000.003C.0180.F800.0000.1137  ;2038  MCT/ALLOW.IB.READ
U 1137. 0000.003C.0180.F800.0000.1138  ;2039  MCT/ALLOW.IB.READ
U 1138. 1000.003C.0180.F800.0000.1139  ;2040  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1139. 0000.003E.0180.0800.0000.0001  ;2041  RETURN1
U 113A. 3000.003C.0180.0800.0000.1133  ;2042 WAITIB: IBC/START,J/IBW
            ;2043 ;
            ;2044 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
            ;2045 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
            ;2046 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
            ;2047 ; RC[8]  A OPERAND BITS <63:32>
            ;2048 ; RC[7]  A OPERAND BITS <31:00>
            ;2049 ; RC[6]  B OPERAND BITS <63:32>
            ;2050 ; RC[5]  B OPERAND BITS <31:00>
            ;2051 ; RC[0A]  RESULT BITS <63:32>
            ;2052 ; RC[9]  RESULT BITS <31:00>
            ;2053 ;
U 113B. 0810.0038.0180.0938.0000.113C  ;2054 ADDSIM: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 113C. 0010.0038.01C0.C928.0000.113D  ;2055  Q_RC[5]
U 113D. 001D.0014.0180.09C8.0010.113E  ;2056  ALU_D+Q,RC[9]_ALU,CLK.UBCC ; AND SAVE THE CARRY
U 113E. 0810.1B38.0180.0940.0000.100E  ;2057  D_RC[8],ALU.C?
U 100E. 0000.003C.0180.0800.0000.113F  ;2058 =1110 J/ADSM1   ; NO CARRY
U 100F. 0819.0014.0580.0800.0000.113F  ;2059  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
            ;2060     ; RESULT
```

```
U 113F, 0010,0038,01C0,0930,0000,1140  ;2061 ADSM1: Q_RC[6]
U 1140, 001D,0016,0180,09D0,0000,0001  ;2062  ALU_D+Q,RC[0A]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
            ;2063      ; AND RETURN.
            ;2064 ;
            ;2065 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU SUBTRACT. IT PERFORMS A
            ;2066 ; DOUBLE PRECISION SUB ON OPERANDS FOUND IN RC[7], RC[8], RC[5]
            ;2067 ; AND RC[6] AND LEAVES THE RESULT IN RC[0A] AND RC[9].
            ;2068 ; RC[8]  A OPERAND BITS <63:32>
            ;2069 ; RC[7]  A OPERAND BITS <31:00>
            ;2070 ; RC[6]  B OPERAND BITS <63:32>
            ;2071 ; RC[5]  B OPERAND BITS <31:00>
            ;2072 ;
U 1141, 0010,0038,01C0,0938,0000,1142  ;2073 SUBSIM: Q_RC[7]   ; DO LO SUB
U 1142, 0810,0038,0180,0928,0000,1143  ;2074  D_RC[5]
U 1143, 001D,0000,0180,09C8,0010,1144  ;2075  ALU_D-Q,RC[9]_ALU,CLK,UBCC ; SAVE BORROW
U 1144, 0010,1B38,01C0,0940,0000,101E  ;2076  Q_RC[8],ALU,C?
U 101E, 0019,2014,05C0,0800,0000,101F  ;2077 =1110 ALU_Q+K[.1],Q_ALU ; SUB BORROW
U 101F, 0810,0038,0180,0930,0000,1145  ;2078 SBSM1: D_RC[6]
U 1145, 001D,0002,0180,09D0,0000,0001  ;2079  ALU_D-Q,RC[0A]_ALU,RETURN1 ; DO UPPER HALF OF SUB.
            ;2080 ;
            ;2081 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
            ;2082 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[7], RC[8], R[0D]
            ;2083 ; AND R[0C] AND LEAVES THE RESULT IN RC[8] AND RC[7].
            ;2084 ; RC[8]  A OPERAND BITS <63:32>
            ;2085 ; RC[7]  A OPERAND BITS <31:00>
            ;2086 ; R[0C]  B OPERAND BITS <63:32>
            ;2087 ; R[0D]  B OPERAND BITS <31:00>
            ;2088 ; RC[8]  RESULT BITS <63:32>
            ;2089 ; RC[7]  RESULT BITS <31:00>
            ;2090 ;
U 1146, 0810,003B,0180,0938,0000,1147  ;2091 INCA: D_RC[7]   ; FIRST ADD LO FRACTIONS
U 1147, 0000,003C,01C0,0A68,0000,1148  ;2092  Q_R[0D]
U 1148, 001D,0014,0180,09B8,0010,1149  ;2093  ALU_D+Q,RC[7]_ALU,CLK,UBCC ; AND SAVE THE CARRY
U 1149, 0810,1B38,0180,0940,0000,102E  ;2094  D_RC[8],ALU,C?
U 102E, 0000,003C,0180,0800,0000,114A  ;2095 =1110 J/INCA1   ; NO CARRY
U 102F, 0819,0014,0580,0800,0000,114A  ;2096  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
            ;2097      ; RESULT
U 114A, 0000,003C,01C0,0A60,0000,114B  ;2098 INCA1: Q_R[0C]
U 114B, 001D,0016,0180,09C0,0000,0001  ;2099  ALU_D+Q,RC[8]_ALU,RETURN1  ; DO HIGH HALF OF THE ADD
            ;2100      ; AND RETURN.
            ;2101 ;
            ;2102 ; THIS SUBROUTINE IS CALLED TO SIMULATE A FALU ADDD. IT PERFORMS A
            ;2103 ; DOUBLE PRECISION ADD ON OPERANDS FOUND IN RC[5], RC[6], R[0D]
            ;2104 ; AND R[0C] AND LEAVES THE RESULT IN RC[6] AND RC[5].
            ;2105 ; RC[6]  A OPERAND BITS <63:32>
            ;2106 ; RC[5]  A OPERAND BITS <31:00>
            ;2107 ; R[0C]  B OPERAND BITS <63:32>
            ;2108 ; R[0D]  B OPERAND BITS <31:00>
            ;2109 ; RC[6]  RESULT BITS <63:32>
            ;2110 ; RC[5]  RESULT BITS <31:00>
            ;2111 ;
U 114C, 0810,0038,0180,0928,0000,114D  ;2112 INCB: D_RC[5]   ; FIRST ADD LO FRACTIONS
U 114D, 0000,003C,01C0,0A68,0000,114E  ;2113  Q_R[0D]
U 114E, 0019,2024,59C0,0800,0000,114F  ;2114  ALU_Q[ANDNOT]K[.7F],Q_ALU
U 114F, 001D,0014,0180,09A8,0010,1150  ;2115  ALU_D+Q,RC[5]_ALU,CLK,UBCC ; AND SAVE THE CARRY
```

```
U 1150, 0810.1B38.0180.0930.0000.103E  ;2116  D_RC[6],ALU.C?
U 103E, 0000.003C.0180.0800.0000.1151  ;2117 =1110 J/INCB1    ; NO CARRY
U 103F, 0819.0014.0580.0800.0000.1151  ;2118  ALU_D+K[.1],D_ALU ; CARRY WAS 1 SO ADD TO HIGH
          ;2119      ; RESULT
U 1151, 0000.003C.01C0.0A60.0000.1152  ;2120 INCB1: Q_R[0C]
U 1152, 001D.0016.0180.09B0.0000.0001  ;2121  ALU_D+Q,RC[6]_ALU,RETURN1   ; DO HIGH HALF OF THE ADD
          ;2122      ; AND RETURN.
          ;2123 ;
          ;2124 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2125 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2126 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2127 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2128 ;
U 1153, 0000.003C.01C0.0A28.0000.1154  ;2129 UNSEQA: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1154, 0800.003C.B980.0A20.0084.7156  ;2130  D_R[4],SC_K[.19]
U 1156, 0D00.003C.0180.0800.0000.1157  ;2131  D_DAL.SC
U 1157, 0000.003C.65E0.0800.0084.7158  ;2132  Q_D,SC_K[.10]
U 1158, 0D00.003C.0180.0800.0000.1159  ;2133  D_DAL.SC
U 1159, 0001.003C.0180.0AB8.0000.115A  ;2134  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 115A, 0800.003C.01C0.0A20.0000.115B  ;2135  D_R[4],Q_R[4]
U 115B, 0819.0024.5980.0800.0000.115C  ;2136  ALU_D[ANDNOT]K[.7F],D_ALU
U 115C, 0100.003C.5D88.0800.0084.715D  ;2137  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 115D, 0D00.003C.0180.0800.0000.115E  ;2138  D_DAL.SC
U 115E, 0001.003C.0180.0AB0.0084.715F  ;2139  R[6]_D,SC_K[.8]
U 115F, 0818.0034.59F8.0A28.0000.1160  ;2140  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 1160, 0D00.003C.01C0.0A30.0000.1161  ;2141  D_DAL.SC,Q_R[6]
U 1161, 001D.0032.0180.0AB0.0000.0001  ;2142  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2143 ;
          ;2144 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
          ;2145 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
          ;2146 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
          ;2147 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
          ;2148 ;
U 1162, 0000.003C.01C0.0A28.0000.1163  ;2149 UNSEQB: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1163, 0800.003C.B980.0A20.0084.7164  ;2150  D_R[4],SC_K[.19]
U 1164, 0D00.003C.0180.0800.0000.1165  ;2151  D_DAL.SC
U 1165, 0000.003C.65E0.0800.0084.7166  ;2152  Q_D,SC_K[.10]
U 1166, 0D00.003C.0180.0800.0000.1167  ;2153  D_DAL.SC
U 1167, 0001.003C.0180.0AB8.0000.1168  ;2154  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 1168, 0800.003C.01C0.0A20.0000.1169  ;2155  D_R[4],Q_R[4]
U 1169, 0819.0024.5980.0800.0000.116A  ;2156  ALU_D[ANDNOT]K[.7F],D_ALU
U 116A, 0100.003C.5D88.0800.0084.716B  ;2157  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 116B, 0D00.003C.0180.0800.0000.116C  ;2158  D_DAL.SC
U 116C, 0001.003C.0180.0AB0.0084.716D  ;2159  R[6]_D,SC_K[.8]
U 116D, 0818.0034.59F8.0A28.0000.116E  ;2160  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 116E, 0D00.003C.01C0.0A30.0000.116F  ;2161  D_DAL.SC,Q_R[6]
U 116F, 001D.0032.0180.0AB0.0000.0001  ;2162  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
          ;2163 ;
          ;2164 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
          ;2165 ; IN RC[0A] AND BITS <31:00> IN RC[9]) INTO THE FLOATING POINT FORMAT
          ;2166 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN RC[0E]
          ;2167 ; AND BITS <31:00> IN RC[0D].
          ;2168 ;
U 1170, 0810.0038.0180.0950.0000.1171  ;2169 UNSRES: D_RC[0A]
U 1171, 0010.0038.01C0.0948.0000.1172  ;2170  Q_RC[9]
```

```
U 1172, 0001.003C.0180.0AA0.0000.1030  ;2171  R[4]_D
U 1030, 0001.203D.0180.0AA8.0000.1153  ;2172 =0 R[5]_Q,CALL,J/UNSEQA
U 1031, 0800.003C.0180.0A30.0000.1173  ;2173  D_R[6]
U 1173, 0000.003C.01C0.0A38.0000.1174  ;2174  Q_R[7]
U 1174, 0001.003C.0180.09F0.0000.1175  ;2175  RC[0E]_D
U 1175, 0001.203E.0180.09E8.0000.0001  ;2176  RC[0D]_Q,RETURN1
           ;2177 ;
           ;2178 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
           ;2179 ; IN RC[8] AND BITS <31:00> IN RC[7]) INTO THE FLOATING POINT FORMAT
           ;2180 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[8]
           ;2181 ; AND BITS <31:00> IN R[9].
           ;2182 ;
U 1176, 0810.0038.0180.0940.0000.1177  ;2183 UNSAOP: D_RC[8]
U 1177, 0010.0038.01C0.0938.0000.1178  ;2184  Q_RC[7]
U 1178, 0001.003C.0180.0AA0.0000.1034  ;2185  R[4]_D
U 1034, 0001.203D.0180.0AA8.0000.1153  ;2186 =0 R[5]_Q,CALL,J/UNSEQA
U 1035, 0800.003C.0180.0A30.0000.1179  ;2187  D_R[6]
U 1179, 0000.003C.01C0.0A38.0000.117A  ;2188  Q_R[7]
U 117A, 0001.003C.0180.0AC0.0000.117B  ;2189  R[8]_D
U 117B, 0001.203E.0180.0AC8.0000.0001  ;2190  R[9]_Q,RETURN1
           ;2191 ;
           ;2192 ; THIS ROUTINE IS CALLED TO PACK A FRACTION (BITS <63:32>
           ;2193 ; IN RC[6] AND BITS <31:00> IN RC[5]) INTO THE FLOATING POINT FORMAT
           ;2194 ; AND LEAVE THE PACKED RESULT BITS <63:32> IN R[0A]
           ;2195 ; AND BITS <31:00> IN R[0B].
           ;2196 ;
U 117C, 0810.0038.0180.0930.0000.117D  ;2197 UNSBOP: D_RC[6]
U 117D, 0010.0038.01C0.0928.0000.117E  ;2198  Q_RC[5]
U 117E, 0001.003C.0180.0AA0.0000.103C  ;2199  R[4]_D
U 103C, 0001.203D.0180.0AA8.0000.1153  ;2200 =0 R[5]_Q,CALL,J/UNSEQA
U 103D, 0800.003C.0180.0A30.0000.117F  ;2201  D_R[6]
U 117F, 0000.003C.01C0.0A38.0000.1180  ;2202  Q_R[7]
U 1180, 0001.003C.0180.0AD0.0000.1181  ;2203  R[0A]_D
U 1181, 0001.203E.0180.0AD8.0000.0001  ;2204  R[0B]_Q,RETURN1
           ;2205 ;
           ;2206 ; THIS ROUTINE IS CALLED BY THE FALU TESTS TO LOAD LA, SA,
           ;2207 ; LB, SB, AR AND BR. FALU/A.B FUNCTION IS EXECUTED
           ;2208 ; AND THE RESULTS RETRIEVED.
           ;2209 ;
U 1182, 0818.0038.3180.0800.0000.1040  ;2210 EXEC: D_K[.40]
U 1040, 0819.0031.6180.0800.0000.111B  ;2211 =0 ALU_D[OR]K[.F],D_ALU,CALL,J/GENTRA
U 1041, 0000.003C.5980.3C00.0000.1183  ;2212  ID[ACC.2]_D
U 1183, 0000.00FC.0380.0800.0000.1184  ;2213  TRAP.FPA  ; TRAP TO FPA 4F
U 1184, 0000.003C.0180.0A10.0000.1185  ;2214  LAB_R[2]  ; AT FPA 4F
U 1185, 0000.007C.0180.0A10.0000.1186  ;2215  LAB_R[2],CPSYNC  ; AT FPA 4F
U 1186, 0000.003C.0180.0A18.0000.1187  ;2216  LAB_R[3]  ; AT FPA 39
U 1187, 0000.007C.0180.0A18.0000.1188  ;2217  LAB_R[3],CPSYNC  ; AT FPA 39
U 1188, 0000.003C.0180.0A40.0000.1189  ;2218  LAB_R[8]  ; AT FPA 3A
U 1189, 0000.007C.0180.0A40.0000.118A  ;2219  LAB_R[8],CPSYNC  ; AT FPA 3A
U 118A, 0000.003C.0180.0A48.0000.118B  ;2220  LAB_R[9]  ; AT FPA 3B
U 118B, 0000.007C.0180.0A48.0000.118C  ;2221  LAB_R[9],CPSYNC  ; AT FPA 3B
U 118C, 0000.003C.0180.0A50.0000.118D  ;2222  LAB_R[0A]  ; AT FPA 3C
U 118D, 0000.007C.0180.0A50.0000.118E  ;2223  LAB_R[0A],CPSYNC ; AT FPA 3C
U 118E, 0000.003C.0180.0A58.0000.118F  ;2224  LAB_R[0B]  ; AT FPA 3D
U 118F, 0000.007C.0180.0A58.0000.1190  ;2225  LAB_R[0B],CPSYNC ; AT FPA 3D
```

```
U 1190, 0A00.003C.0180.0800.0000.1191  ;2226  D_ACC   ; AT FPA 3E, GET RESULT HI
U 1191, 0000.003C.01D8.0800.0000.1192  ;2227  Q_ACC   ; AT FPA 3F, GET RESULT LO
U 1192, 0001.003C.0180.09E0.0000.1193  ;2228  RC[0C]_D
U 1193, 0001.203E.0180.09D8.0000.0001  ;2229  RC[0B]_Q,RETURN1
            ;2230 ;*
            ;2231 ; THE FOLLOWING ROUTINES READ THE FPA STATUS REGISTER AND
            ;2232 ; DETERMINE WHETHER BIT 31 (ERROR SUMMARY BIT) AND/OR BIT
            ;2233 ; 27 (RSV OPERAND) ARE ASSERTED OR DEASSERTED.
            ;2234 ;-
            ;2235
            ;2236
            ;2237
            ;2238 ; BTS 31, 27 ASSERTED?
            ;2239
U 1194, 0000.003C.5DF0.2C00.0000.1195  ;2240 ACCHK1: Q_ID[ACC.CS]
U 1195, 0001.203C.0180.09E8.0000.1196  ;2241  RC[0D]_Q
U 1196, 0000.003C.0180.0800.0000.1197  ;2242 ACCHK2: NOP
U 1197, 0000.0D3C.0180.0800.0000.1003  ;2243  Q31?
U 1003, 0000.003E.0180.0800.0000.0001  ;2244 =011 RETURN1
U 1007, 0000.003C.0188.0800.0000.1198  ;2245  Q_Q.LEFT2
U 1198, 0000.003C.0188.0800.0000.1199  ;2246  Q_Q.LEFT2
U 1199, 0000.0D3C.0180.0800.0000.1013  ;2247  Q31?
U 1013, 0000.003E.0180.0800.0000.0001  ;2248 =011 RETURN1
U 1017, 0810.0038.0180.0948.0000.119A  ;2249  D_RC[9]
U 119A, 0001.003C.0180.09F0.0000.119B  ;2250  RC[0E]_D  ; SAVE EXPECTED
U 119B, 0000.003E.0180.0800.0000.0002  ;2251  RETURN2
            ;2252
            ;2253
            ;2254
            ;2255
            ;2256 ; BITS 31, 27 NOT ASSERTED?
            ;2257
U 119C, 0000.003C.5DF0.2C00.0000.119D  ;2258 ACCHK3: Q_ID[ACC.CS]
U 119D, 0000.0D3C.0180.0800.0000.1023  ;2259  Q31?
U 1023, 0000.003C.0180.0800.0000.119E  ;2260 =011 J/ACC1
U 1027, 0000.003E.0180.0800.0000.0001  ;2261  RETURN1
U 119E, 0000.003C.0188.0800.0000.119F  ;2262 ACC1: Q_Q.LEFT2
U 119F, 0000.003C.0188.0800.0000.11A0  ;2263  Q_Q.LEFT2
U 11A0, 0000.0D3C.0180.0800.0000.1033  ;2264  Q31?
U 1033, 0000.003C.0180.0800.0000.11A1  ;2265 =011 J/ACC2
U 1037, 0000.003E.0180.0800.0000.0001  ;2266  RETURN1
U 11A1, 0818.0038.4580.0800.0000.11A2  ;2267 ACC2: D_K[.8000]
U 11A2, 0019.0014.0580.09F0.0000.11A3  ;2268  ALU_D+K[.1],RC[0E]_ALU  ; SAVE EXPECTED
U 11A3, 0000.003E.0180.0800.0000.0002  ;2269  RETURN2
            ;2270
            ;2271
            ;2272 ; BIT 31 ASSERTED, 27 NOT ASSERTED?
            ;2273
U 11A4, 0000.003C.5DF0.2C00.0000.11A5  ;2274 ACCHK4: Q_ID[ACC.CS]
U 11A5, 0001.203C.0180.09E8.0000.11A6  ;2275  RC[0D]_Q
U 11A6, 0000.0D3C.0180.0800.0000.1043  ;2276  Q31?
U 1043, 0000.003E.0180.0800.0000.0001  ;2277 =011 RETURN1
U 1047, 0000.003C.0188.0800.0000.11A7  ;2278  Q_Q.LEFT2
U 11A7, 0000.003C.0188.0800.0000.11A8  ;2279  Q_Q.LEFT2
U 11A8, 0000.0D3C.0180.0800.0000.104B  ;2280  Q31?
```

```
U 104B, 0000.003C.0180.0800.0000.11A9  ;2281 =011 J/ACC4
U 104F, 0000.003E.0180.0800.0000.0001  ;2282  RETURN1
U 11A9, 0810.0038.0180.0950.0000.11AA  ;2283 ACC4: D_RC[0A]
U 11AA, 0001.003C.0180.09F0.0000.11AB  ;2284  RC[0E]_D  ; SAVE EXPECTED
U 11AB, 0000.003E.0180.0800.0000.0002  ;2285  RETURN2
            ;2286
            ;2287
            ;2288 ;*
            ;2289 ; THIS ROUTINE LOADS THE PR IN THE EXPONENT PROCESSOR, FROM R5.
            ;2290 ; THE PR IS ROUTED TO THE EALU OUTPUT VIA THE AMX. THE PR PROVIDES
            ;2291 ; A ZERO EXPONENT AT THE EALU OUTPUT. THIS WILL ASSERT FORCE ZERO.
            ;2292 ;-
            ;2293
U 11AC, 0818.0038.E980.0800.0000.11AD  ;2294 LD.PR: D_K[.24]
U 11AD, 0000.003C.0180.0800.0000.1044  ;2295  NOP
U 1044, 0819.0015.0580.0800.0000.111B  ;2296 =0 D_D+K[.1],CALL[GENTRA]
U 1045, 0000.003C.5980.3C00.0000.11AE  ;2297  ID[ACC.2]_D
U 11AE, 0000.00FC.0380.0A28.0000.11AF  ;2298  TRAP.FPA,LAB_R[5]
U 11AF, 0000.007C.0180.0A28.0000.11B0  ;2299  LAB_R[5],CPSYNC ; FPA AT 025. LOAD LA
U 11B0, 0000.003C.0180.0800.0000.11B1  ;2300  NOP  ; FPA AT 026. LOAD PR
U 11B1, 0000.003E.0180.0800.0000.0001  ;2301  RETURN1
            ;2302
            ;2303
            ;2304 ;*
            ;2305 ; THIS ROUTINE TRAPS TO FPA ADDRESS 087. AT THIS ADDRESS SA AND LA
            ;2306 ; DATA IS LOADED FROM R5. THE FPA ADDRESS IS THE ADVANCED TO 088
            ;2307 ; WHERE SB AND LB DATA IS LOADED FROM R6. R5 AND R6 ARE LOADED
            ;2308 ; IN THE CALLING PROGRAM.
            ;2309 ;-
            ;2310
            ;2311 =0
U 1048, 0000.003D.0180.0800.0000.11B6  ;2312 LD.SEX: CALL[TRP.87]
U 1049, 0000.003C.0180.0A28.0000.11B2  ;2313  LAB_R[5]  ; FPA AT 087.
U 11B2, 0000.007C.0180.0A28.0000.11B3  ;2314  LAB_R[5],CPSYNC  ; LOAD SA, LA.
U 11B3, 0000.003C.0180.0A30.0000.11B4  ;2315  LAB_R[6]  ; FPA AT 088.
U 11B4, 0000.007C.0180.0A30.0000.11B5  ;2316  LAB_R[6],CPSYNC  ; LOAD SB, LB.
U 11B5, 0000.003E.0180.0800.0000.0001  ;2317  RETURN1  ; FPA AT 028.
            ;2318
            ;2319
U 11B6, 0818.0038.4180.0800.0000.104C  ;2320 TRP.87: D_K[.80]
U 104C, 0819.0031.5D80.0800.0000.111B  ;2321 =0 D_D.OR.K[.7],CALL[GENTRA]
U 104D, 0000.003C.5980.3C00.0000.11B7  ;2322  ID[ACC.2]_D
U 11B7, 0000.00FC.0380.0800.0000.11B8  ;2323  TRAP.FPA
U 11B8, 0000.003E.0180.0800.0000.0001  ;2324  RETURN1
            ;2325
            ;2326 ; THESE ADDRESSES ARE RESERVED FOR THE WCS DEBUGGER.
            ;2327
U 1120, 0000.003C.0180.0800.0000.1121  ;2328 1120: NOP
U 1121, 0000.003C.0180.0800.0000.1122  ;2329 1121: NOP
U 1122, 0000.003C.0180.0800.0000.1123  ;2330 1122: NOP
U 1123, 0000.003C.0180.0800.0000.1124  ;2331 1123: NOP
U 1124, 0000.003C.0180.0800.0000.1125  ;2332 1124: NOP
U 1125, 0000.003C.0180.0800.0000.1126  ;2333 1125: NOP
U 1126, 0000.003C.0180.0800.0000.1127  ;2334 1126: NOP
U 1127, 0000.003C.0180.0800.0000.1128  ;2335 1127: NOP
```

```
U 1128, 0000,003C,0180,0800,0000,1129  ;2336 1128: NOP
U 1129, 0000,003C,0180,0800,0000,112A  ;2337 1129: NOP
U 112A, 0000,003C,0180,0800,0000,112B  ;2338 112A: NOP
U 112B, 0000,003C,0180,0800,0000,112C  ;2339 112B: NOP
U 112C, 0000,003C,0180,0800,0000,112D  ;2340 112C: NOP
U 112D, 0000,003C,0180,0800,0000,112E  ;2341 112D: NOP
U 112E, 0000,003C,0180,0800,0000,112F  ;2342 112E: NOP
U 112F, 0000,003C,0180,0800,0000,1058  ;2343 112F: NOP
            ;2344
            ;2345
            ;2346
            ;2347
            ;2348
            ;2349
            ;2350
```

```
;2351  .PAGE "TEST 277 COMPLEMENT AR TEST"
;2352 ;*****************************************************************
;2353 ;**
;2354 ; COMPLEMENT AR TEST
;2355 ;
;2356 ; TEST DESCRIPTION
;2357 ; THIS IS A TEST OF THE COMPLEMENT AR FUNCTION. A 1 IS FLOATED
;2358 ; ACROSS AR AND EXPONENT VALUES ARE SET UP ALONG WITH THE SIGNS
;2359 ; WHICH SHOULD CAUSE THE COMPLEMENT AR TO TAKE PLACE.
;2360 ;
;2361 ; FPA UCODE USED
;2362 ; LOCATION CONTENTS
;2363 ; 68:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/69
;2364 ; 69:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/6A
;2365 ; 6A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/6C
;2366 ; 6C:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/6D
;2367 ; 6D:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/83
;2368 ; 83:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/PSTALL
;2369 ; 55:  BSC/NH,EALUC/K3FF,FADC/BR0,NAD/55
;2370 ; 53:  FADC/A.B,BSC/FAL.X,NAD/178
;2371 ; 178:  WAIT,NAD/0FB
;2372 ; 0FB:  BSC/FAL.HL,FADC/A,WAIT,NAD/0FC
;2373 ; 0FC:  BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/0FD
;2374 ; 0FD:  BSC/FAL.HL,FADC/B,WAIT,FPSYNC,NAD/0FE
;2375 ; 0FE:  BSC/FAL.LH,FADC/B,WAIT,FPSYNC,NAD/0FE
;2376 ;
;2377 ; ERROR DESCRIPTION
;2378 ; EXPECTED AR HI, AFTER COMPLEMENT
;2379 ; EXPECTED AR LO, AFTER COMPLEMENT
;2380 ; GOT AR HI
;2381 ; GOT AR LO
;2382 ; EXPECTED AR HI BEFORE COMPLEMENT, IN FRACTION FORM
;2383 ; EXPECTED AR LO BEFORE COMPLEMENT, IN FRACTION FORM
;2384 ;
;2385 ;--
;2386 ;*****************************************************************
;2387
;2388 ; NOTE: THE DATA FOR THIS TEST IS FOUND AT THE END OF TEST 255.
;2389
;2390 =0
U 1058, 0000.003D.0180.0800.0000.104A  ;2391 CAR1: NEWTST
U 1059, 0018.0038.D580.09F8.0000.11B9  ;2392  RC[0F]_K[.6]
U 11B9, 0818.0038.7980.0800.0000.11BA  ;2393  D_K[.30]  ; GENERATE EXPONENTS
U 11BA, 0819.0030.D980.0800.0000.11BB  ;2394  ALU_D[OR]K[.9],D_ALU
U 11BB, 0000.003C.5DF8.0800.0084.71BC  ;2395  SC_K[.7],Q_0
U 11BC, 0D00.003C.0180.0800.0000.11BD  ;2396  D_DAL.SC
U 11BD, 0019.0030.4580.0A88.0000.11BE  ;2397  ALU_D[OR]K[.8000],R[1]_ALU ; LB=39, SB=1
U 11BE, 0018.0038.4180.0A90.0000.11BF  ;2398  R[2]_K[.80]  ; LA=1, SA=0
U 11BF, 0018.0038.F5C0.0800.0000.11C0  ;2399  Q_K[.A]  ; GENERATE BR VALUE
U 11C0, 0019.2030.25C0.0800.0000.11C1  ;2400  ALU_Q[OR]K[.A0],Q_ALU
U 11C1, 0F00.003C.7180.0800.0084.71C2  ;2401  SC_K[.FFF8],D_0
U 11C2, 0D00.003C.0180.0800.0000.11C3  ;2402  D_DAL.SC
U 11C3, 0D00.003C.0180.0800.0000.11C4  ;2403  D_DAL.SC
U 11C4, 0D00.003C.0180.0800.0000.11C5  ;2404  D_DAL.SC
U 11C5, 0D00.003C.0180.0800.0000.11C6  ;2405  D_DAL.SC
```

```
U 11C6. 0001.003C.0180.0AE0.0000.11C7  ;2406  R[0C]_D
U 11C7. 0001.003C.0180.0AE8.0000.11C8  ;2407  R[0D]_D
U 11C8. 0818.0038.3180.0800.0000.105C  ;2408  D_K[.40]  ; GENERATE ADDRESSES OF FPA UCODE
U 105C. 0819.0031.2D80.0800.0000.111B  ;2409 =0 ALU_D[OR]K[.28],D_ALU,CALL,J/GENTRA
U 105D. 0001.003C.0180.0AF0.0000.11C9  ;2410  R[0E]_D
U 11C9. 0818.0038.3580.0800.0000.11CA  ;2411  D_K[.50]
U 11CA. 0819.0030.8980.0800.0000.106C  ;2412  ALU_D[OR]K[.D],D_ALU
U 106C. 0000.003D.0180.0800.0000.111B  ;2413 =0 CALL,J/GENTRA
U 106D. 0001.003C.0180.0AF8.0000.11CB  ;2414  R[0F]_D
U 11CB. 0818.0038.3580.0800.0000.1070  ;2415  D_K[.50]
U 1070. 0819.0015.0D80.0800.0000.111B  ;2416 =0 ALU_D+K[.3],D_ALU,CALL,J/GENTRA
U 1071. 0001.003C.0180.0A98.0000.11CC  ;2417  R[3]_D
U 11CC. 0818.0038.3580.0800.0000.11CD  ;2418  D_K[.50]
U 11CD. 0819.0014.0580.0800.0000.1072  ;2419  ALU_D+K[.1],D_ALU
U 1072. 0819.0015.1180.0800.0000.111B  ;2420 =0 ALU_D+K[.4],D_ALU,CALL,J/GENTRA
U 1073. 0001.003C.0180.0980.0000.11CE  ;2421  RC[0]_D
U 11CE. 0018.0038.5580.0A80.0000.1074  ;2422  R[0]_K[.3F]  ; SET UP A COUNTER
U 1074. 0000.003D.0180.0800.0000.10FF  ;2423 =0 ERLOOP
U 1075. 0000.003C.0180.0800.0000.1076  ;2424  NOP
          ;2425 =0
U 1076. 0000.003D.0180.0800.0000.1116  ;2426 CAR1A: CALL,J/FPINIT
U 1077. 0018.0038.7580.0800.0200.1078  ;2427  VA_K[.20]  ; LOAD ADDD INTO IB
U 1078. 0000.003D.0180.0800.0000.1131  ;2428 =0 CALL,J/LOADIB
U 1079. 0800.003C.0180.0A00.0082.11CF  ;2429  D_R[0],SC_R[0]  ; GENERATE AROPERAND.
U 11CF. 0019.0034.7580.0800.0010.11D0  ;2430  ALU_D[AND]K[.20],CLK.UBCC
U 11D0. 0000.013C.0180.0800.0000.107A  ;2431  Z?
U 107A. 0000.003C.0180.0800.0000.11D2  ;2432 =0 J/CAR1B
U 107B. 0003.003C.0180.0AA0.0000.11D1  ;2433  R[4]_0
U 11D1. 0003.001C.0180.0AA8.0000.11D4  ;2434  ALU_NOT.MASK,R[5]_ALU,J/CAR1C
U 11D2. 0003.003C.0180.0AA8.0000.11D3  ;2435 CAR1B: R[5]_0
U 11D3. 0003.001C.0180.0AA0.0000.11D4  ;2436  ALU_NOT.MASK,R[4]_ALU
U 11D4. 0800.003C.0180.0A20.0000.11D5  ;2437 CAR1C: D_R[4]
U 11D5. 0000.003C.01C0.0A28.0000.11D6  ;2438  Q_R[5]
U 11D6. 0001.003C.0180.09D0.0000.107C  ;2439  RC[0A]_D
U 107C. 0001.203D.0180.09C8.0000.1153  ;2440 =0 RC[9]_Q,CALL,J/UNSEQA
U 107D. 0800.003C.0180.0A30.0000.11D7  ;2441  D_R[6]
U 11D7. 0000.003C.01C0.0A38.0000.11D8  ;2442  Q_R[7]
U 11D8. 0001.003C.0180.0AC0.0000.11D9  ;2443  R[8]_D
U 11D9. 0001.203C.0180.0AC8.0000.11DA  ;2444  R[9]_Q
U 11DA. 0000.003C.01F8.0800.0000.11DB  ;2445  Q_0  ; GENERATE EXPECTED RESULT
U 11DB. 0800.003C.0180.0A28.0000.11DC  ;2446  D_R[5]  ; OF FIRST COMPLEMENT FUNCTION.
U 11DC. 001D.2000.0180.0AA8.0010.11DD  ;2447  ALU_Q-D,R[5]_ALU,CLK.UBCC
U 11DD. 0800.1B3C.0180.0A20.0000.106E  ;2448  ALU.C?,D_R[4]
U 106E. 001B.0000.05C0.0800.0000.106F  ;2449 =1110 ALU_0-K[.1],Q_ALU
U 106F. 001D.2000.0180.0AA0.0000.1080  ;2450 CAR1D: ALU_Q-D,R[4]_ALU
U 1080. 0000.003D.0180.0800.0000.1153  ;2451 =0 CALL,J/UNSEQA
U 1081. 0800.003C.0180.0A30.0000.11DE  ;2452  D_R[6]
U 11DE. 0000.003C.01C0.0A38.0000.11DF  ;2453  Q_R[7]
U 11DF. 0001.003C.0180.09F0.0000.11E0  ;2454  RC[0E]_D
U 11E0. 0001.203C.0180.09E8.0000.11E1  ;2455  RC[0D]_Q
U 11E1. 0800.003C.0180.0A70.0000.11E2  ;2456  D_R[0E]
U 11E2. 0000.003C.5980.3C00.0000.11E3  ;2457  ID[ACC.2]_D
U 11E3. 0000.00FC.0380.0800.0000.11E4  ;2458  TRAP.FPA  ; TRAP TO FPA 68
U 11E4. 0000.003C.0180.0A10.0000.11E5  ;2459  LAB_R[2]  ; AT FPA 68
U 11E5. 0000.007C.0180.0A10.0000.11E6  ;2460  LAB_R[2],CPSYNC  ; AT FPA 68
```

```
U 11E6. 0000.003C.0180.0A08.0000.11E7  ;2461  LAB_R[1]  ; AT FPA 69
U 11E7. 0000.007C.0180.0A08.0000.11E8  ;2462  LAB_R[1],CPSYNC  ; AT FPA 69
U 11E8. 0000.003C.0180.0A40.0000.11E9  ;2463  LAB_R[8]  ; AT FPA 6A
U 11E9. 0000.007C.0180.0A40.0000.11EA  ;2464  LAB_R[8],CPSYNC  ; AT FPA 6A
U 11EA. 0000.003C.0180.0A48.0000.11EB  ;2465  LAB_R[9]  ; AT FPA 6C
U 11EB. 0000.007C.0180.0A48.0000.11EC  ;2466  LAB_R[9],CPSYNC  ; AT FPA 6C
U 11EC. 0000.003C.0180.0A60.0000.11ED  ;2467  LAB_R[0C]  ; AT FPA 6D
U 11ED. 0000.007C.0180.0A60.0000.11EE  ;2468  LAB_R[0C],CPSYNC ; AT FPA 6D
U 11EE. 0000.003C.0180.0A60.0000.11EF  ;2469  LAB_R[0C]  ; AT FPA 83
U 11EF. 0000.007C.0180.0A60.0000.11F0  ;2470  LAB_R[0C],CPSYNC ; AT FPA 83
U 11F0. 0810.0038.0180.0900.0000.11F1  ;2471  D_RC[0]
U 11F1. 0000.003C.5980.3C00.0000.11F2  ;2472  ID[ACC.2]_D
U 11F2. 0000.00FC.0380.0800.0000.11F3  ;2473  TRAP.FPA
U 11F3. 0800.003C.0180.0A18.0000.11F4  ;2474  D_R[3]  ; AT FPA 55
U 11F4. 0000.003C.5980.3C00.0000.11F5  ;2475  ID[ACC.2]_D
U 11F5. 0000.00FC.0380.0800.0000.11F6  ;2476  TRAP.FPA  ; TRAP TO FPA 53
U 11F6. 0000.003C.0180.0800.0000.11F7  ;2477  NOP  ; AT FPA 53
U 11F7. 0000.003C.0180.0800.0000.11F8  ;2478  NOP  ; AT FPA 178
U 11F8. 0000.007C.0180.0800.0000.11F9  ;2479  NOP,CPSYNC  ; AT FPA 178
U 11F9. 0A00.007C.0180.0800.0000.11FA  ;2480  D_ACC,CPSYNC  ; AT FPA FB, GET AR HI
U 11FA. 0000.003C.01D8.0800.0000.11FB  ;2481  Q_ACC  ; AT FPA FC, GET AR LO
U 11FB. 0001.003C.0180.09E0.0000.11FC  ;2482  RC[0C]_D
U 11FC. 0001.203C.0180.09D8.0000.11FD  ;2483  RC[0B]_Q
U 11FD. 0010.0038.01C0.0970.0000.11FE  ;2484  Q_RC[0E]
U 11FE. 001D.0000.0180.0800.0010.11FF  ;2485  ALU_D-Q,CLK.UBCC ; CHECK AR HI
U 11FF. 0000.013C.0180.0800.0000.1082  ;2486  Z?
U 1082. 0000.003D.0180.0800.0000.110B  ;2487 =0 ERROR2  ; AR HI INCORRECT
U 1083. 0810.0038.0180.0958.0000.1200  ;2488  D_RC[0B]
U 1200. 0010.0038.01C0.0968.0000.1201  ;2489  Q_RC[0D]
U 1201. 001D.0000.0180.0800.0010.1202  ;2490  ALU_D-Q,CLK.UBCC ; CHECK AR LO
U 1202. 0000.013C.0180.0800.0000.1084  ;2491  Z?
U 1084. 0000.003D.0180.0800.0000.110B  ;2492 =0 ERROR2  ; AR LO INCORRECT
U 1085. 0000.003C.0180.0A00.0010.1203  ;2493  ALU_R[0],CLK.UBCC ; TRIED ALL PATTERNS?
U 1203. 0000.013C.0180.0800.0000.1086  ;2494  Z?
U 1086. 0018.0000.0580.0A80.0000.1076  ;2495 =0 R[0]_LA-K[.1],J/CAR1A ; IF NOT CONTINUE
U 1087. 0000.003C.0180.0800.0000.1088  ;2496  NOP
          ;2497 ;
```

```
            ;2498  .PAGE "TEST 278 COMPLEMENT BR TEST"
            ;2499 ;*****************************************************************
            ;2500 ;**
            ;2501 ; COMPLEMENT BR TEST
            ;2502 ;
            ;2503 ; TEST DESCRIPTION
            ;2504 ; THIS IS A TEST OF THE COMPLEMENT BR FUNCTION. A 1 IS FLOATED
            ;2505 ; ACROSS BR AND EXPONENT VALUES ARE SET UP ALONG WITH THE SIGNS
            ;2506 ; WHICH SHOULD CAUSE THE COMPLEMENT BR TO TAKE PLACE.
            ;2507 ;
            ;2508 ; FPA UCODE USED
            ;2509 ; LOCATION CONTENTS
            ;2510 ; 68:  BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/69
            ;2511 ; 69:  BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/6A
            ;2512 ; 6A:  BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/6C
            ;2513 ; 6C:  BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/6D
            ;2514 ; 6D:  BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/83
            ;2515 ; 83:  BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/PSTALL
            ;2516 ; 8E:  BSC/NH,EALUC/K3FF,FADC/AR0,NAD/8E
            ;2517 ; 53:  FADC/A.B,BSC/FAL.X,NAD/178
            ;2518 ; 178:  WAIT,NAD/0FB
            ;2519 ; 0FB:  BSC/FAL.HL,FADC/A,WAIT,NAD/0FC
            ;2520 ; 0FC:  BSC/FAL.LH,FADC/A,WAIT,FPSYNC,NAD/0FD
            ;2521 ; 0FD:  BSC/FAL.HL,FADC/B,WAIT,FPSYNC,NAD/0FE
            ;2522 ; 0FE:  BSC/FAL.LH,FADC/B,WAIT,FPSYNC,NAD/0FE
            ;2523 ;
            ;2524 ; ERROR DESCRIPTION
            ;2525 ; EXPECTED BR HI, AFTER COMPLEMENT
            ;2526 ; EXPECTED BR LO, AFTER COMPLEMENT
            ;2527 ; GOT BR HI
            ;2528 ; GOT BR LO
            ;2529 ; EXPECTED BR HI BEFORE COMPLEMENT, IN FRACTION FORM
            ;2530 ; EXPECTED BR LO BEFORE COMPLEMENT, IN FRACTION FORM
            ;2531 ;
            ;2532 ;--
            ;2533 ;*****************************************************************
            ;2534 =0
U 1088,  0000.003D.0180.0800.0000.104A  ;2535 CBR1: NEWTST
U 1089,  0018.0038.D580.09F8.0000.1204  ;2536  RC[0F]_K[.6]
U 1204,  0818.0038.7980.0800.0000.1205  ;2537  D_K[.30]  ; GENERATE EXPONENTS
U 1205,  0819.0030.D980.0800.0000.1206  ;2538  ALU_D[OR]K[.9],D_ALU
U 1206,  0000.003C.5DF8.0800.0084.7207  ;2539  SC_K[.7],Q_0
U 1207,  0D00.003C.0180.0800.0000.1208  ;2540  D_DAL.SC
U 1208,  0019.0030.4580.0A88.0000.1209  ;2541  ALU_D[OR]K[.8000],R[1]_ALU ; LA=39, SA=1
U 1209,  0018.0038.4180.0A90.0000.120A  ;2542  R[2]_K[.80]  ; LB=1, SB=0
U 120A,  0018.0038.F5C0.0800.0000.120B  ;2543  Q_K[.A]  ; GENERATE AR VALUE
U 120B,  0019.2030.25C0.0800.0000.120C  ;2544  ALU_Q[OR]K[.A0],Q_ALU
U 120C,  0F00.003C.7180.0800.0084.720D  ;2545  SC_K[.FFF8],D_0
U 120D,  0D00.003C.0180.0800.0000.120E  ;2546  D_DAL.SC
U 120E,  0D00.003C.0180.0800.0000.120F  ;2547  D_DAL.SC
U 120F,  0D00.003C.0180.0800.0000.1210  ;2548  D_DAL.SC
U 1210,  0D00.003C.0180.0800.0000.1211  ;2549  D_DAL.SC
U 1211,  0001.003C.0180.0AE0.0000.1212  ;2550  R[0C]_D
U 1212,  0001.003C.0180.0AE8.0000.1213  ;2551  R[0D]_D
U 1213,  0818.0038.3180.0800.0000.108A  ;2552  D_K[.40]  ; GENERATE ADDRESSES OF FPA UCODE
```

```
U 108A. 0819.0031.2D80.0800.0000.111B  ;2553 =0 ALU_D[OR]K[.28],D_ALU,CALL,J/GENTRA
U 108B. 0001.003C.0180.0AF0.0000.1214  ;2554  R[0E]_D
U 1214. 0818.0038.3580.0800.0000.108C  ;2555  D_K[.50]
U 108C. 0819.0015.0D80.0800.0000.111B  ;2556 =0 ALU_D+K[.3],D_ALU,CALL,J/GENTRA
U 108D. 0001.003C.0180.0AF8.0000.1215  ;2557  R[0F]_D
U 1215. 0818.0038.C580.0800.0000.108E  ;2558  D_K[.88]
U 108E. 0819.0031.D580.0800.0000.111B  ;2559 =0 ALU_D[OR]K[.6],D_ALU,CALL,J/GENTRA
U 108F. 0001.003C.0180.0980.0000.1217  ;2560  RC[0]_D
U 1217. 0018.0038.5580.0A80.0000.1090  ;2561  R[0]_K[.3F]  ; SET UP A COUNTER
U 1090. 0000.003D.0180.0800.0000.10FF  ;2562 =0 ERLOOP
U 1091. 0000.003C.0180.0800.0000.1092  ;2563  NOP
        ;2564 =0
U 1092. 0000.003D.0180.0800.0000.1116  ;2565 CBR1A: CALL,J/FPINIT
U 1093. 0018.0038.7580.0800.0200.1094  ;2566  VA_K[.20]  ; LOAD ADDD INTO IB
U 1094. 0000.003D.0180.0800.0000.1131  ;2567 =0 CALL,J/LOADIB
U 1095. 0800.003C.0180.0A00.0082.1218  ;2568  D_R[0],SC_R[0]  ; GENERATE BR OPERAND.
U 1218. 0019.0034.7580.0800.0010.1219  ;2569  ALU_D[AND]K[.20],CLK.UBCC
U 1219. 0000.013C.0180.0800.0000.1096  ;2570  Z?
U 1096. 0000.003C.0180.0800.0000.121B  ;2571 =0 J/CBR1B
U 1097. 0003.003C.0180.0AA0.0000.121A  ;2572  R[4]_0
U 121A. 0003.001C.0180.0AA8.0000.121D  ;2573  ALU_NOT.MASK,R[5]_ALU,J/CBR1C
U 121B. 0003.003C.0180.0AA8.0000.121C  ;2574 CBR1B: R[5]_0
U 121C. 0003.001C.0180.0AA0.0000.121D  ;2575  ALU_NOT.MASK,R[4]_ALU
U 121D. 0800.003C.0180.0A20.0000.121E  ;2576 CBR1C: D_R[4]
U 121E. 0000.003C.01C0.0A28.0000.121F  ;2577  Q_R[5]
U 121F. 0019.2024.59C0.0AA8.0000.1220  ;2578  ALU_Q[ANDNOT]K[.7F],Q_ALU,R[5]_ALU
U 1220. 0001.003C.0180.09D0.0000.1098  ;2579  RC[0A]_D
U 1098. 0001.203D.0180.09C8.0000.1162  ;2580 =0 RC[9]_Q,CALL,J/UNSEQB
U 1099. 0800.003C.0180.0A30.0000.1221  ;2581  D_R[6]
U 1221. 0000.003C.01C0.0A38.0000.1222  ;2582  Q_R[7]
U 1222. 0001.003C.0180.0AC0.0000.1223  ;2583  R[8]_D
U 1223. 0001.203C.0180.0AC8.0000.1224  ;2584  R[9]_Q
U 1224. 0000.003C.01F8.0800.0000.1225  ;2585  Q_0  ; GENERATE EXPECTED RESULT
U 1225. 0800.003C.0180.0A28.0000.1226  ;2586  D_R[5]  ; OF FIRST COMPLEMENT FUNCTION.
U 1226. 001D.2000.0180.0AA8.0010.1227  ;2587  ALU_Q-D,R[5]_ALU,CLK.UBCC
U 1227. 0800.1B3C.0180.0A20.0000.107E  ;2588  ALU.C?,D_R[4]
U 107E. 001B.0000.05C0.0800.0000.107F  ;2589 =1110 ALU_0-K[.1],Q_ALU
U 107F. 001D.2000.0180.0AA0.0000.109A  ;2590 CBR1D: ALU_Q-D,R[4]_ALU
U 109A. 0000.003D.0180.0800.0000.1162  ;2591 =0 CALL,J/UNSEQB
U 109B. 0800.003C.0180.0A30.0000.1228  ;2592  D_R[6]
U 1228. 0000.003C.01C0.0A38.0000.1229  ;2593  Q_R[7]
U 1229. 0001.003C.0180.09F0.0000.122A  ;2594  RC[0E]_D
U 122A. 0001.203C.0180.09E8.0000.122B  ;2595  RC[0D]_Q
U 122B. 0800.003C.0180.0A70.0000.122C  ;2596  D_R[0E]
U 122C. 0000.003C.5980.3C00.0000.122D  ;2597  ID[ACC.2]_D
U 122D. 0000.00FC.0380.0800.0000.122E  ;2598  TRAP.FPA  ; TRAP TO FPA 68
U 122E. 0000.003C.0180.0A08.0000.122F  ;2599  LAB_R[1]  ; AT FPA 68
U 122F. 0000.007C.0180.0A08.0000.1230  ;2600  LAB_R[1],CPSYNC  ; AT FPA 68
U 1230. 0000.003C.0180.0A10.0000.1231  ;2601  LAB_R[2]  ; AT FPA 69
U 1231. 0000.007C.0180.0A10.0000.1232  ;2602  LAB_R[2],CPSYNC  ; AT FPA 69
U 1232. 0000.003C.0180.0A60.0000.1233  ;2603  LAB_R[0C]  ; AT FPA 6A
U 1233. 0000.007C.0180.0A60.0000.1234  ;2604  LAB_R[0C],CPSYNC ; AT FPA 6A
U 1234. 0000.003C.0180.0A60.0000.1235  ;2605  LAB_R[0C]  ; AT FPA 6C
U 1235. 0000.007C.0180.0A60.0000.1236  ;2606  LAB_R[0C],CPSYNC ; AT FPA 6C
U 1236. 0000.003C.0180.0A40.0000.1237  ;2607  LAB_R[8]  ; AT FPA 6D
```

```
U 1237. 0000.007C.0180.0A40.0000.1238  ;2608  LAB_R[8].CPSYNC  ; AT FPA 6D
U 1238. 0000.003C.0180.0A48.0000.1239  ;2609  LAB_R[9]  ; AT FPA 83
U 1239. 0000.007C.0180.0A48.0000.123A  ;2610  LAB_R[9].CPSYNC  ; AT FPA 83
U 123A. 0810.0038.0180.0900.0000.123B  ;2611  D_RC[0]
U 123B. 0000.003C.5980.3C00.0000.123C  ;2612  ID[ACC.2]_D
U 123C. 0000.00FC.0380.0800.0000.123D  ;2613  TRAP.FPA  ; TRAP TO FPA 8E
U 123D. 0800.003C.0180.0A78.0000.123E  ;2614  D_R[0F]
U 123E. 0000.003C.5980.3C00.0000.123F  ;2615  ID[ACC.2]_D
U 123F. 0000.00FC.0380.0800.0000.1240  ;2616  TRAP.FPA  ; TRAP TO FPA 53
U 1240. 0000.003C.0180.0800.0000.1241  ;2617  NOP  ; AT FPA 53, BR COMPLEMENTED HERE
U 1241. 0000.003C.0180.0800.0000.1242  ;2618  NOP  ; AT FPA 178
U 1242. 0000.007C.0180.0800.0000.1243  ;2619  NOP.CPSYNC  ; AT FPA 178
U 1243. 0000.007C.0180.0800.0000.1244  ;2620  NOP.CPSYNC  ; AT FPA 0FB
U 1244. 0000.007C.0180.0800.0000.1245  ;2621  NOP.CPSYNC  ; AT FPA 0FC
U 1245. 0A00.007C.0180.0800.0000.1246  ;2622  D_ACC.CPSYNC  ; AT FPA 0FD, GET BR HI
U 1246. 0000.003C.01D8.0800.0000.1247  ;2623  Q_ACC  ; AT FPA 0FE, GET BR LO
U 1247. 0001.003C.0180.09E0.0000.1248  ;2624  RC[0C]_D
U 1248. 0001.203C.0180.09D8.0000.1249  ;2625  RC[0B]_Q
U 1249. 0010.0038.01C0.0970.0000.124A  ;2626  Q_RC[0E]
U 124A. 001D.0000.0180.0800.0010.124B  ;2627  ALU_D-Q.CLK.UBCC ; CHECK BR HI
U 124B. 0000.013C.0180.0800.0000.109C  ;2628  Z?
U 109C. 0000.003D.0180.0800.0000.110B  ;2629 =0 ERROR2  ; BR HI INCORRECT
U 109D. 0810.0038.0180.0958.0000.124C  ;2630  D_RC[0B]
U 124C. 0010.0038.01C0.0968.0000.124D  ;2631  Q_RC[0D]
U 124D. 001D.0000.0180.0800.0010.124E  ;2632  ALU_D-Q.CLK.UBCC ; CHECK BR LO
U 124E. 0000.013C.0180.0800.0000.109E  ;2633  Z?
U 109E. 0000.003D.0180.0800.0000.110B  ;2634 =0 ERROR2  ; BR LO INCORRECT
U 109F. 0000.003C.0180.0A00.0010.124F  ;2635  ALU_R[0].CLK.UBCC ; TRIED ALL PATTERNS?
U 124F. 0000.013C.0180.0800.0000.10A0  ;2636  Z?
U 10A0. 0018.0000.0580.0A80.0000.1092  ;2637 =0 R[0]_LA-K[.1],J/CBR1A ; IF NOT CONTINUE
U 10A1. 0000.003C.0180.0800.0000.10A2  ;2638  NOP
        ;2639
        ;2640
```

```
;2641   .PAGE "TEST 279 TEST FPA ERROR LATCH AND CPU BEN/6."
;2642 ;*****************************************************************
;2643 ;++
;2644 ; TEST FPA ERROR LATCH AND CPU BEN/6.
;2645 ;
;2646 ; TEST DESCRIPTION
;2647 ;
;2648 ; THIS TEST EXERCISES THE LOGIC THAT SETS AND RESETS THE
;2649 ; FPA ERRPR LATCH. THE LATCH IS VERIFIED BY CHECKING THE FPA
;2650 ; STATUS REGISTER BIT 31 (ERROR SUMMARY) AND BIT 27 (RSV OPERAND)
;2651 ; THE CPU BEN/6 BRANCH CONDITIONS ARE ALSO TESTED.
;2652 ;
;2653 ; FPA UCODE USED
;2654 ; LOCATION CONTENTS
;2655 ; 028 NAD/028
;2656 ; 000: NAD/0A3,EALUC/B,FADC/LD,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
;2657 ; 0A3: SCR/R.R,FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
;2658 ; 087: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,FADC/R1,WAIT,NAD/028
;2659 ; 088: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,FADC/BR1,WAIT,NAD/028
;2660 ; 0B5: FADC/A,BSC/FAL.HL,NRC/LD,BEN/3,NAD/104
;2661 ; 104: MSC/M0,NAD/107
;2662 ; 107: MSC/M0,NAD/104
;2663 ; 01F: BSC/R,SCR/CPU,EAC,LDAB,NAD/06B,WAIT
;2664 ; 025: BSC/R,SCR/CPU,EAC/LDA,NAD/026,WAIT
;2665 ; 026: AMXC/LA,EALUC/A,EAC/LDA,NAD/027
;2666 ; 0F3: BSC/NH,SGNC/A.RES,MSC/1,NAD/028
;2667 ; 0C1: BSC/PQ,NRC/LD,MSC/RR.0,NAD/133
;2668 ; 133: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC/CC,EALU/A+B,AMXC/PR,
;2669 ;  BMXC/NK,FPSYNC,BEN/4,NAD/0A4
;2670 ; 0A4: BMXC/NK,BSC/NH,AMXC/PR,EALUC/A,FPSYNC,WAIT,BEN/4,NAD/0A6
;2671 ; 0A6: BMXC/NK,BSC/NL,AMXC/PR,EALUC/A,FPSYNC,WAIT,NAD/IRD
;2672 ;
;2673 ;
;2674 ;
;2675 ; LOGIC DESCRIPTION
;2676 ;
;2677 ; THE ERROR LATCH, BEN3 LOGIC IS ON THE FCT MODULE. THE STATUS
;2678 ; REGISTER LOGIC IS ON THE FMH MODULE.
;2679 ;
;2680 ; ERROR DESCRIPTION
;2681 ;
;2682 ; EXPECTED STATUS REGISTER
;2683 ; RECEIVED STATUS REGISTER
;2684 ; EXPECTED BEN/6
;2685 ; RECEIVED BEN/6
;2686 ;
;2687 ;
;2688 ; SYNC POINT DESCRIPTION
;2689 ;
;2690 ;--
;2691 ;*****************************************************************
;2692
;2693
;2694 =0
U 10A2, 0018,0039,1180,09F8,0000,104A  ;2695 ERRCHK: NEWTST[.4]
```

```
U 10A3, 0000.003C.8D80.0800.0084.7250  ;2696  SC_K[.1F]
U 1250, 0803.001C.0180.0800.0000.1251  ;2697  D_NOT.MASK
U 1251, 0819.0014.0580.0800.0000.1252  ;2698  ALU_D+K[.1],D_ALU
U 1252, 0019.0030.4580.09D0.0000.1253  ;2699  RC[0A]_D.OR.K[.8000] ; ASSERT BITS 31,15,1
U 1253, 0000.003C.ED80.0800.0084.7254  ;2700  SC_K[.1B]
U 1254, 0803.001C.0180.0800.0000.1255  ;2701  D_NOT.MASK
U 1255, 0010.0038.01C0.0950.0000.1256  ;2702  Q_RC[0A]
U 1256, 001D.0030.0180.09C8.0000.10A4  ;2703  ALU_D.OR.Q,RC[9]_ALU ; ASSERT BITS 31 27,15,1
U 10A4, 0000.003D.0180.0800.0000.10FF  ;2704 =0 ERLOOP
U 10A5, 0000.003C.0180.0800.0000.10A6  ;2705  NOP
U 10A6, 0000.003D.0180.0800.0000.1116  ;2706 =0 CALL[FPINIT]  ; IB GETS HALT INSTRUCTION
          ;2707
          ;2708
          ;2709 ;+
          ;2710 ; ZERO FPA STATUS REGISTER. CHECK STATUS REGISTER BITS 31 27.
          ;2711 ; TEST BEN/6 100
          ;2712 ;-
          ;2713
U 10A7, 0818.0038.4580.0800.0000.1257  ;2714  D_K[.8000]
U 1257, 0000.003C.5D80.3C00.0000.10A8  ;2715  ID[ACC.CS]_D  ; ZERO STATUS REGISTER AND ERROR LATCH
U 10A8, 0818.0639.1180.0800.0000.1050  ;2716 =0 BEN/6,CALL,J/BEN6,D_K[.4]
U 10A9, 0001.003C.0180.09E0.0000.1258  ;2717  RC[0C]_D   ; SAVE EXPECTED
U 1258, 0001.203C.0180.09D8.0000.1259  ;2718  RC[0B]_Q   ; SAVE RECEIVED
U 1259, 001D.2000.0180.0800.0010.125A  ;2719  ALU_Q-D,CLK.UBCC
U 125A, 0000.013C.0180.0800.0000.10AA  ;2720  Z?
U 10AA, 0000.003D.0180.0800.0000.1109  ;2721 =0 ERROR1
U 10AB, 0000.003C.0180.0800.0000.1008  ;2722  NOP
          ;2723
          ;2724 ; CHECK STATUS REGISTER ERROR BITS
          ;2725
U 1008, 0000.003D.0180.0800.0000.119C  ;2726 =00 CALL[ACCHK3]  ; CHECK BITS 31, 27 NOT ASSERTED
U 1009, 0000.003D.0180.0800.0000.1109  ;2727  ERROR1
U 100A, 0000.003C.0180.0800.0000.125B  ;2728  J/STAT1   ; BITS 31, 27 NOT ASSERTED
          ;2729 =     ; TERMINATE UPC ADDRESS CONSTRAINT
          ;2730
          ;2731
          ;2732 ;+
          ;2733 ; THIS IS THE BEN6 TABLE FOR: -ACC UB2--ACC UB1--ACC UB0-
          ;2734 ;       OPCODE BIT 7,ACC. BIT 31,FPSYNC.
          ;2735 ;-
          ;2736
          ;2737 =000
U 1050, 0000.003E.01F8.0800.0000.0001  ;2738 BEN6: Q_0,RETURN1
U 1051, 0018.003A.05C0.0800.0000.0001  ;2739  Q_K[.1],RETURN1
U 1052, 0018.003A.09C0.0800.0000.0001  ;2740  Q_K[.2],RETURN1
U 1053, 0018.003A.0DC0.0800.0000.0001  ;2741  Q_K[.3],RETURN1
U 1054, 0018.003A.11C0.0800.0000.0001  ;2742  Q_K[.4],RETURN1
U 1055, 001B.0012.11C0.0800.0000.0001  ;2743  ALU_0+K[.4]+1,Q_ALU,RETURN1
U 1056, 0018.003A.D5C0.0800.0000.0001  ;2744  Q_K[.6],RETURN1
U 1057, 0018.003A.5DC0.0800.0000.0001  ;2745  Q_K[.7],RETURN1
          ;2746
          ;2747 ; TEST BEN/6 000
          ;2748
U 125B, 0018.0038.0180.0800.0200.10AC  ;2749 STAT1: VA_K[.8]
U 10AC, 0000.003D.0180.0800.0000.1131  ;2750 =0 CALL[LOADIB]   ; NULL R,R. ASSERT BEN/6 ACC UB2
```

```
ZZ-ESKAR-V22  TEST 279 TEST FPA ERROR LATCH AND CPU BEN/6.                                    SEQ 2613
ESKAR60.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:14                        Page   6

U 10AD. 0000.003C.0180.0800.0000.10AE  ;2751  NOP
U 10AE. 0000.003D.0180.0800.0000.10FF  ;2752 =0 ERLOOP
U 10AF. 0000.003C.0180.0800.0000.10B0  ;2753  NOP
U 10B0. 0F00.063D.0180.0800.0000.1050  ;2754 =0 BEN/6,CALL,J/BEN6,D_0
U 10B1. 0001.003C.0180.09E0.0000.125C  ;2755  RC[0C]_D
U 125C. 0001.203C.0180.09D8.0000.125D  ;2756  RC[0B]_Q
U 125D. 001D.2000.0180.0800.0010.125E  ;2757  ALU_Q-D,CLK.UBCC
U 125E. 0000.013C.0180.0800.0000.10B2  ;2758  Z?
U 10B2. 0000.003D.0180.0800.0000.1109  ;2759 =0 ERROR1
           ;2760
           ;2761
           ;2762 ;*
           ;2763 ; CHECK FM0 (FLOATING POINT MINUS ZERO) INPUT TO THE  ERROR LATCH
           ;2764 ; TEST BEN/6 010
           ;2765 ;_
           ;2766
U 10B3. 0018.0038.4580.0AA8.0000.125F  ;2767  R[5]_K[.8000]
U 125F. 0003.003C.0180.0AB0.0000.10B4  ;2768  R[6]_0
U 10B4. 0000.003D.0180.0800.0000.1048  ;2769 =0 CALL[LD.SEX]  ; SA_1,SB_0,LA_0,LB_0
           ;2770     ; FORCE RSV OPERAND
           ;2771
           ;2772 ; TRAP TO 0B5 TO ASSERT FPA BEN 3
           ;2773
U 10B5. 0000.003C.0180.0800.0000.10B6  ;2774  NOP
U 10B6. 0000.003D.0180.0800.0000.10FF  ;2775 =0 ERLOOP
U 10B7. 0818.0038.9580.0800.0000.10B8  ;2776  D_K[.B0]
U 10B8. 0819.0011.1180.0800.0000.111B  ;2777 =0 ALU_D+K[.4]+1,D_ALU,CALL[GENTRA]
U 10B9. 0000.003C.5980.3C00.0000.1260  ;2778  ID[ACC.2]_D
U 1260. 0000.00FC.0380.0800.0000.1261  ;2779  TRAP.FPA
U 1261. 0000.003C.0180.0800.0000.1262  ;2780  NOP    ; FPA AT 0B5.
U 1262. 0000.003C.0180.0800.0000.1263  ;2781  NOP    ; FPA AT 104.
U 1263. 0000.003C.5DF0.2C00.0000.1264  ;2782  Q_ID[ACC.CS]   ; FPA AT 104. READ FM0
U 1264. 0001.203C.0180.09E8.0000.10BA  ;2783  RC[0D]_Q   ; FPA AT 104. SAVE
U 10BA. 0818.0639.0980.0800.0000.1050  ;2784 =0 BEN/6,CALL,J/BEN6,D_K[.2] ; FPA AT 104. ASSERT RSV (MSC/2)
U 10BB. 0001.003C.0180.09E0.0000.1265  ;2785  RC[0C]_D   ; SAVE EXPECTED
U 1265. 0001.203C.0180.09D8.0000.1266  ;2786  RC[0B]_Q   ; SAVE RECEIVED
U 1266. 001D.2000.0180.0800.0010.1267  ;2787  ALU_Q-D,CLK.UBCC
U 1267. 0000.013C.0180.0800.0000.10BC  ;2788  Z?
U 10BC. 0000.003D.0180.0800.0000.1109  ;2789 =0 ERROR1
U 10BD. 0010.0038.01C0.0968.0000.1018  ;2790  Q_RC[0D]   ; LOAD Q FOR STATUS BITS CHECK
           ;2791
           ;2792
           ;2793 ; CHECK STATUS REGISTER BITS
           ;2794
U 1018. 0000.003D.0180.0800.0000.1196  ;2795 =00 CALL[ACCHK2]   ; CHECK BITS 31, 27 ASSERTED
U 1019. 0000.003D.0180.0800.0000.1109  ;2796  ERROR1
U 101A. 0000.003C.0180.0800.0000.10BE  ;2797  NOP
           ;2798 =    ; TERMINATE UPC ADDRESS CONSTRAINT
           ;2799
           ;2800
           ;2801
           ;2802 ;*
           ;2803 ; ISSUE IRD STALL TO CHECK ERROR LATCH RESET
           ;2804 ; TEST BEN/6 100
           ;2805 ;_
```

```
            ;2806
U 10BE, 0000.003D.0180.0800.0000.10FF  ;2807 =0 ERLOOP
U 10BF, 0000.003C.0180.0800.0000.10C0  ;2808  NOP
U 10C0, 0F00.003D.0180.0800.0000.111B  ;2809 =0 D_0,CALL[GENTRA]
U 10C1, 0000.003C.5980.3C00.0000.1268  ;2810  ID[ACC.2]_D
U 1268, 0000.00FC.0380.0800.0000.1269  ;2811  TRAP.FPA
U 1269, 0000.003C.0180.0800.0000.126A  ;2812  NOP    ; FPA AT 000.
U 126A, 0000.003C.0180.0800.0000.126B  ;2813  NOP    ; FPA AT 0A3. (IRD)
U 126B, 0018.0038.8580.0800.0200.10C2  ;2814  VA_K[.C]
U 10C2, 0000.003D.0180.0800.0000.1131  ;2815 =0 CALL[LOADIB]   ; IB_ACCELERATER INSTRUCTION.
U 10C3, 0000.003C.0180.0800.0000.1028  ;2816  NOP
U 1028, 0000.003D.0180.0800.0000.1029  ;2817 =00 CALL
U 1029, 0000.003F.0180.0800.0000.1216  ;2818  SUB/SPEC,J/IRTAB3
U 102A, 0000.003C.0180.0800.0000.102B  ;2819  NOP
U 102B, 0000.003C.0180.0800.0000.10C4  ;2820  NOP
U 10C4, 0818.0639.1180.0800.0000.1050  ;2821 =0 BEN/6,CALL,J/BEN6,D_K[.4]
U 10C5, 0001.203C.0180.09D8.0000.126C  ;2822  RC[0B]_Q   ; SAVE RECEIVED
U 126C, 001D.2000.0180.0800.0010.126D  ;2823  ALU_Q-D,CLK.UBCC
U 126D, 0000.013C.0180.0800.0000.10C6  ;2824  Z?
U 10C6, 0000.003D.0180.0800.0000.1109  ;2825 =0 ERROR1
U 10C7, 0000.003C.0180.0800.0000.1038  ;2826  NOP
            ;2827
            ;2828 ; CHECK STATUS REGISTER ERROR BITS
            ;2829
U 1038, 0000.003D.0180.0800.0000.119C  ;2830 =00 CALL[ACCHK3]  ; CHECK BITS 31, 27 NOT ASSERTED
U 1039, 0000.003D.0180.0800.0000.1109  ;2831  ERROR1
U 103A, 0000.003C.0180.0800.0000.10C8  ;2832  NOP    ; BITS 31,27 NOT ASSERTED
            ;2833 =     ; TERMINATE UPC ADDRESS CONSTRAINT
            ;2834
            ;2835 ;*
            ;2836 ; CHECK (FORCE ZERO*(-ERROR CHKL+POLYADDL)) INPUTS TO THE ERROR LATCH
            ;2837 ; TEST BEN/6 110
            ;2838 ;_
            ;2839
U 10C8, 0000.003D.0180.0800.0000.10FF  ;2840 =0 ERLOOP
U 10C9, 0000.003C.0180.0800.0000.126E  ;2841  NOP
U 126E, 0003.003C.0180.0AA8.0000.126F  ;2842  R[5]_0
U 126F, 0003.003C.0180.0AB0.0000.10CA  ;2843  R[6]_0
U 10CA, 0000.003D.0180.0800.0000.11AC  ;2844 =0 CALL[LD.PR]  ; LA_0,PR_0,LB_0
U 10CB, 0000.003C.0180.0800.0000.10CC  ;2845  NOP    ; ASSERT FORCE ZERO
            ;2846
            ;2847 ; TRAP TO 0F3 TO ASSERT POLYADDL.
            ;2848
U 10CC, 0000.003D.0180.0800.0000.1116  ;2849 =0 CALL[FPINIT]   ; IB_HALT INSTRUCTION
U 10CD, 0818.0038.3D80.0800.0000.10CE  ;2850  D_K[.EF]
U 10CE, 0819.0015.1180.0800.0000.111B  ;2851 =0 ALU_D+K[.4],D_ALU,CALL[GENTRA]
U 10CF, 0000.003C.5980.3C00.0000.1270  ;2852  ID[ACC.2]_D
U 1270, 0000.00FC.0380.0800.0000.1271  ;2853  TRAP.FPA
U 1271, 0000.003C.0180.0800.0000.1272  ;2854  NOP    ; FPA AT 0F3. POLY ADDL ASSERTED (MSC/1)
U 1272, 0000.003C.0180.0800.0000.10D0  ;2855  NOP    ; FPA AT 028.
            ;2856
            ;2857 ; CHECK FOR ERROR SUMMARY BIT SET.
            ;2858
U 10D0, 0818.0639.D580.0800.0000.1050  ;2859 =0 BEN/6,CALL,J/BEN6,D_K[.6]
U 10D1, 0001.003C.0180.09E0.0000.1273  ;2860  RC[0C]_D   ; SAVE EXPECTED
```

```
U 1273, 0001,203C,0180,09D8,0000,1274  ;2861  RC[0B]_Q   ; SAVE RECEIVED
U 1274, 0010,0038,01C0,0958,0000,1275  ;2862  Q_RC[0B]   ; RESTORE Q REGISTER
U 1275, 001D,2000,0180,0800,0010,1276  ;2863  ALU_Q-D,CLK.UBCC
U 1276, 0000,013C,0180,0800,0000,10D2  ;2864  Z?
U 10D2, 0000,003D,0180,0800,0000,1109  ;2865 =0 ERROR1
U 10D3, 0000,003C,0180,0800,0000,1060  ;2866  NOP
          ;2867
          ;2868 ; CHECK STATUS REGISTER ERROR BITS.
          ;2869
U 1060, 0000,003D,0180,0800,0000,11A4  ;2870 =00 CALL[ACCHK4] ; CHECK BIT 31 ASSERTED, BIT 27 NOT ASSERTED
U 1061, 0000,003D,0180,0800,0000,1109  ;2871  ERROR1
U 1062, 0000,003C,0180,0800,0000,10D4  ;2872  NOP
          ;2873 =    ; TERMINATE UPC ADDRESS CONSTRAINT
          ;2874
          ;2875 ;*
          ;2876 ; CHECK (FORCE ZERO*(ERROR CHKL+(-POLYADDL)))
          ;2877 ; TEST BEN/6 011
          ;2878 ;_
          ;2879
U 10D4, 0000,003D,0180,0800,0000,10FF  ;2880 =0 ERLOOP
U 10D5, 0000,003C,0180,0800,0000,1277  ;2881  NOP
U 1277, 0818,0038,4580,0800,0000,1278  ;2882  D_K[.8000]
U 1278, 0000,003C,5D80,3C00,0000,1279  ;2883  ID[ACC.CS]_D   ; RESET ERROR LATCH
U 1279, 0003,003C,0180,0AA8,0000,127A  ;2884  R[5]_0
U 127A, 0003,003C,0180,0AB0,0000,10D6  ;2885  R[6]_0
U 10D6, 0000,003D,0180,0800,0000,11AC  ;2886 =0 CALL[LD.PR]   ; LA_0,PR_0,LB_0
          ;2887      ; ASSERT FORCE ZERO
          ;2888
          ;2889
          ;2890 ; TRAP TO 0C1 THEN TO 133 TO ASSERT ERROR CHKL (MSC/CC).
          ;2891
U 10D7, 0018,0038,0180,0800,0200,10D8  ;2892  VA_K[.8]
U 10D8, 0000,003D,0180,0800,0000,1131  ;2893 =0 CALL[LOADIB]   ; MULLR,R
U 10D9, 0818,0038,D180,0800,0000,10DA  ;2894  D_K[.C0]
U 10DA, 0819,0015,0580,0800,0000,111B  ;2895 =0 ALU_D+K[.1],D_ALU,CALL[GENTRA]
U 10DB, 0000,003C,5980,3C00,0000,127B  ;2896  ID[ACC.2]_D
U 127B, 0000,00FC,0380,0800,0000,127C  ;2897  TRAP.FPA
U 127C, 0000,003C,0180,0800,0000,127D  ;2898  NOP   ; FPA AT 0C1.
U 127D, 0000,003C,0180,0800,0000,127E  ;2899  NOP   ; FPA AT 133. MSC/CC
U 127E, 0000,003C,0180,0800,0000,10DC  ;2900  NOP   ; FPA AT 0A4. FPSYNC
U 10DC, 0818,0639,0D80,0800,0000,1050  ;2901 =0 BEN/6,CALL,J/BEN6,D_K[.3]
U 10DD, 0001,003C,0180,09E0,0000,127F  ;2902  RC[0C]_D   ; SAVE EXPECTED
U 127F, 0001,203C,0180,09D8,0000,1280  ;2903  RC[0B]_Q   ; SAVE RECEIVED
U 1280, 001D,2000,0180,0800,0010,1281  ;2904  ALU_Q-D,CLK.UBCC
U 1281, 0000,013C,0180,0800,0000,10DE  ;2905  Z?
U 10DE, 0000,003D,0180,0800,0000,1109  ;2906 =0 ERROR1
U 10DF, 0000,003C,0180,0800,0000,1064  ;2907  NOP
          ;2908
          ;2909 ; CHECK STATUS REGISTER ERROR BIT ASSERTED.
          ;2910
U 1064, 0000,003D,0180,0800,0000,11A4  ;2911 =00 CALL[ACCHK4]
U 1065, 0000,003D,0180,0800,0000,1109  ;2912  ERROR1
U 1066, 0000,003C,0180,0800,0000,10E0  ;2913  NOP   ; BIT 31 ASSERTED, BIT 27 NOT ASSERTED
          ;2914 =    ; TERMINATE UPC ADDRESS CONSTRAINT
          ;2915
```

```
            ;2916
            ;2917
            ;2918 ;*
            ;2919 ; CHECK (-FORCE ZERO*(ERROR CHKL+(-POLYADDL)))
            ;2920 ; TEST BEN/6 001
            ;2921 ;_
            ;2922
U 10E0. 0000.003D.0180.0800.0000.10FF  ;2923 =0 ERLOOP
U 10E1. 0000.003C.0180.0800.0000.1282  ;2924  NOP
U 1282. 0818.0038.4580.0800.0000.1283  ;2925  D_K[.8000]
U 1283. 0000.003C.5D80.3C00.0000.1284  ;2926  ID[ACC.CS]_D   ; RESET ERROR LATCH
U 1284. 0018.0038.8180.0AA8.0000.1285  ;2927  R[5]_K[.3FF]
U 1285. 0018.0038.8180.0AB0.0000.10E2  ;2928  R[6]_K[.3FF]
U 10E2. 0000.003D.0180.0800.0000.11AC  ;2929 =0 CALL[LD.PR]   ; LA>0,PR>0,LB>0
            ;2930      ; DEASSERT FORCE ZERO
            ;2931
            ;2932
            ;2933 ; TRAP TO 0C1 THEN TO 133 TO ASSERT ERROR CHKL (MSC/CC).
            ;2934
U 10E3. 0018.0038.0180.0800.0200.10E4  ;2935  VA_K[.8]
U 10E4. 0000.003D.0180.0800.0000.1131  ;2936 =0 CALL[LOADIB]   ; MULLR,R
U 10E5. 0818.0038.D180.0800.0000.10E6  ;2937  D_K[.C0]
U 10E6. 0819.0015.0580.0800.0000.111B  ;2938 =0 ALU_D+K[.1],D_ALU,CALL[GENTRA]
U 10E7. 0000.003C.5980.3C00.0000.1286  ;2939  ID[ACC.2]_D
U 1286. 0000.00FC.0380.0800.0000.1287  ;2940  TRAP.FPA
U 1287. 0000.003C.0180.0800.0000.1288  ;2941  NOP   ; FPA AT 0C1.
U 1288. 0000.003C.0180.0800.0000.1289  ;2942  NOP   ; FPA AT 133. MSC/CC
U 1289. 0000.003C.0180.0800.0000.10E8  ;2943  NOP   ; FPA AT 0A4. FPSYNC
U 10E8. 0818.0639.0580.0800.0000.1050  ;2944 =0 BEN/6,CALL,J/BEN6,D_K[.1]
U 10E9. 0001.003C.0180.09E0.0000.128A  ;2945  RC[0C]_D   ; SAVE EXPECTED
U 128A. 0001.203C.0180.09D8.0000.128B  ;2946  RC[0B]_Q   ; SAVE RECEIVED
U 128B. 001D.2000.0180.0800.0010.128C  ;2947  ALU_Q-D,CLK.UBCC
U 128C. 0000.013C.0180.0800.0000.10EA  ;2948  Z?
U 10EA. 0000.003D.0180.0800.0000.1109  ;2949 =0 ERROR1
U 10EB. 0000.003C.0180.0800.0000.1068  ;2950  NOP
            ;2951
            ;2952 ; CHECK STATUS REGISTER ERROR BIT DEASSERTED.
            ;2953
U 1068. 0000.003D.0180.0800.0000.119C  ;2954 =00 CALL[ACCHK3]
U 1069. 0000.003D.0180.0800.0000.1109  ;2955  ERROR1
U 106A. 0000.003C.0180.0800.0000.128E  ;2956  J/ENDTST  ; BIT 31 DEASSERTED, BIT 27 NOT ASSERTED
            ;2957 =     ; TERMINATE UPC ADDRESS CONSTRAINT
            ;2958
            ;2959
            ;2960 1216:
U 1216. 0000.003E.0180.0800.0000.0002  ;2961 IRTAB3: RETURN2
U 128D. 0000.003C.0180.0800.0000.128E  ;2962  NOP
            ;2963
            ;2964
            ;2965 ;X8
            ;2966 ;$ 5FC4, 005E
            ;2967 ;$ 5F60, 005F
            ;2968
            ;2969 ;X20
            ;2970 ;$ 5060,0050
```

```
            ;2971
U 128E, 0000,003C,0180,0800,0000,10EC  ;2972 ENDTST: NOP
            ;2973
```

```
ZZ-ESKAR-V22  TEST 27A RESERVED                                                    SEQ 2618
ESKAR60.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:14              Page   7

            ;2974  .PAGE "TEST 27A RESERVED"
            ;2975 =0
U 10EC, 0000,003D,0180,0800,0000,104A  ;2976 DX1: NEWTST
U 10ED, 0000,003C,0180,0800,0000,10EE  ;2977  NOP
            ;2978
```

```
ZZ-ESKAR-V22  TEST 27B RESERVED                                                    SEQ 2619
ESKAR60.MCR                        MICRO2  1P(00)    26-JUL-85  17:05:14          Page   7

            ;2979  .PAGE "TEST 27B RESERVED"
            ;2980 =0
U 10EE, 0000,003D,0180,0800,0000,104A  ;2981 DX2: NEWTST
U 10EF, 0000,003C,0180,0800,0000,10F0  ;2982  NOP
            ;2983
```

```
            ;2984  .PAGE "TEST 27C RESERVED"
            ;2985 =0
U 10F0. 0000.003D.0180.0800.0000.104A  ;2986 DX3: NEWTST
U 10F1. 0000.003C.0180.0800.0000.10F2  ;2987  NOP
            ;2988
```

```
            ;2989  .PAGE "DUMMY NEWTST"
            ;2990 =0
U 10F2, 0000,003D,0180,0800,0000,104A  ;2991 DX5: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 2622
ESKAR60.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:14           Page   8
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1875 2244= 1881= 1897= 1876 2245=
U 1008  2726= 2727= 2728= 1877 1898= 1899= 2058= 2059=
U 1010  1918= 1919= 1878 2248= 1995= 1996= 1879 2249=
U 1018  2795= 2796= 2797= 1880 1998= 1999= 2077= 2078=
U 1020  2000= 2001= 1900 2260= 2003= 2004= 1901 2261=
U 1028  2817= 2818= 2819= 2820= 2023= 2024= 2095= 2096=
U 1030  2172= 2173= 1902 2265= 2186= 2187= 1903 2266=
U 1038  2830= 2831= 2832= 1904 2200= 2201= 2117= 2118=
U 1040  2211= 2212= 1905 2277= 2296= 2297= 1906 2278=
U 1048  2312= 2313= 1916 2281= 2321= 2322= 1939: 2282=
U 1050  2738= 2739= 2740= 2741= 2742= 2743= 2744= 2745=
U 1058  2391= 2392= 1942: 1917 2409= 2410= 1945: 1920
U 1060  2870= 2871= 2872= 1921 2911= 2912= 2913= 1922
U 1068  2954= 2955= 2956= 1923 2413= 2414= 2449= 2450=
U 1070  2416= 2417= 2420= 2421= 2423= 2424= 2426= 2427=
U 1078  2428= 2429= 2432= 2433= 2440= 2441= 2589= 2590=
U 1080  2451= 2452= 2487= 2488= 2492= 2493= 2495= 2496=
U 1088  2535= 2536= 2553= 2554= 2556= 2557= 2559= 2560=
U 1090  2562= 2563= 2565= 2566= 2567= 2568= 2571= 2572=
U 1098  2580= 2581= 2591= 2592= 2629= 2630= 2634= 2635=
U 10A0  2637= 2638= 2695= 2696= 2704= 2705= 2706= 2714=
U 10A8  2716= 2717= 2721= 2722= 2750= 2751= 2752= 2753=
U 10B0  2754= 2755= 2759= 2767= 2769= 2774= 2775= 2776=
U 10B8  2777= 2778= 2784= 2785= 2789= 2790= 2807= 2808=
U 10C0  2809= 2810= 2815= 2816= 2821= 2822= 2825= 2826=
U 10C8  2840= 2841= 2844= 2845= 2849= 2850= 2851= 2852=
U 10D0  2859= 2860= 2865= 2866= 2880= 2881= 2886= 2892=
U 10D8  2893= 2894= 2895= 2896= 2901= 2902= 2906= 2907=
U 10E0  2923= 2924= 2929= 2935= 2936= 2937= 2938= 2939=
U 10E8  2944= 2945= 2949= 2950= 2976= 2977= 2981= 2982=
U 10F0  2986= 2987= 2991= 1924 1925 1926 1927 1928
U 10F8  1929 1930 1931 1932 1933 1934 1935 1936
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1937 1938 1940 1870: 1871: 1872: 1873:
U 1110  1941 1973 1974 1980 1981 1982 1993 1994
U 1118  1997 2002 2005 2015 2016 2017 2025 2026
U 1120  2328: 2329: 2330: 2331: 2332: 2333: 2334: 2335:
U 1128  2336: 2337: 2338: 2339: 2340: 2341: 2342: 2343:
U 1130  2027 2033 2034 2035 2036 2037 2038 2039
U 1138  2040 2041 2042 2054 2055 2056 2057 2061
U 1140  2062 2073 2074 2075 2076 2079 2091 2092
U 1148  2093 2094 2098 2099 2112 2113 2114 2115
U 1150  2116 2120 2121 2129 2130 1965: 2131 2132
U 1158  2133 2134 2135 2136 2137 2138 2139 2140
U 1160  2141 2142 2149 2150 2151 2152 2153 2154
U 1168  2155 2156 2157 2158 2159 2160 2161 2162
U 1170  2169 2170 2171 2174 2175 2176 2183 2184
U 1178  2185 2188 2189 2190 2197 2198 2199 2202
U 1180  2203 2204 2210 2213 2214 2215 2216 2217
U 1188  2218 2219 2220 2221 2222 2223 2224 2225
U 1190  2226 2227 2228 2229 2240 2241 2242 2243
U 1198  2246 2247 2250 2251 2258 2259 2262 2263
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0   2264 2267 2268 2269 2274 2275 2276 2279
U 11A8   2280 2283 2284 2285 2294 2295 2298 2299
U 11B0   2300 2301 2314 2315 2316 2317 2320 2323
U 11B8   2324 2393 2394 2395 2396 2397 2398 2399
U 11C0   2400 2401 2402 2403 2404 2405 2406 2407
U 11C8   2408 2411 2412 2415 2418 2419 2422 2430
U 11D0   2431 2434 2435 2436 2437 2438 2439 2442
U 11D8   2443 2444 2445 2446 2447 2448 2453 2454
U 11E0   2455 2456 2457 2458 2459 2460 2461 2462
U 11E8   2463 2464 2465 2466 2467 2468 2469 2470
U 11F0   2471 2472 2473 2474 2475 2476 2477 2478
U 11F8   2479 2480 2481 2482 2483 2484 2485 2486
U 1200   2489 2490 2491 2494 2537 2538 2539 2540
U 1208   2541 2542 2543 2544 2545 2546 2547 2548
U 1210   2549 2550 2551 2552 2555 2558 2961: 2561
U 1218   2569 2570 2573 2574 2575 2576 2577 2578
U 1220   2579 2582 2583 2584 2585 2586 2587 2588
U 1228   2593 2594 2595 2596 2597 2598 2599 2600
U 1230   2601 2602 2603 2604 2605 2606 2607 2608
U 1238   2609 2610 2611 2612 2613 2614 2615 2616
U 1240   2617 2618 2619 2620 2621 2622 2623 2624
U 1248   2625 2626 2627 2628 2631 2632 2633 2636
U 1250   2697 2698 2699 2700 2701 2702 2703 2715
U 1258   2718 2719 2720 2749 2756 2757 2758 2768
U 1260   2779 2780 2781 2782 2783 2786 2787 2788
U 1268   2811 2812 2813 2814 2823 2824 2842 2843
U 1270   2853 2854 2855 2861 2862 2863 2864 2882
U 1278   2883 2884 2885 2897 2898 2899 2900 2903
U 1280   2904 2905 2925 2926 2927 2928 2940 2941
U 1288   2942 2943 2946 2947 2948 2962 2972
U 1290 - 12A7 Unused
U 12A8     1966:
U 12B0 - 13EF Unused
U 13F0   1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8   1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2624
ESKAR60.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:14     Page   8

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR60        672

Used               672
Remaining

Total microwords used in memory U: 672
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2625
ESKAR60.MCR                    MICRO2  1P(00)    26-JUL-85  17:05:14     Page   8


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR60.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
   ESKAR61.MCR   MICRO2 1P(00)   26-JUL-85 17 SEQ 2626

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2267 TEST 27D SIGN LOGIC FOR MUL/DIV
; 2455 TEST 27E SIGN LOGIC FOR EXACT ZERO, UFLO, OVFLO
; 2638 TEST 27F SIGN LOGIC FOR ADD/SUB WITH LA>LB
; 2885 TEST 280 RESERVED
; 2890 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807   .PAGE "MODULE REVISION HISTORY"
;1808  ;*****************************************************************************
;1809  ;
;1810  ;
;1811  ; MODULE NAME: ESKAR61
;1812  ;
;1813  ; CREATION DATE:  SEPTEMBER 1982
;1814  ; AUTHOR:  DAN GRIGORE
;1815  ;
;1816  ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817  ;
;1818  ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819  ;
;1820  ; -  WHAT CHANGE WAS MADE
;1821  ;
;1822  ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823  ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824  ;
;1825  ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826  ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827  ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828  ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829  ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830  ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831  ;    ESKAR3C.
;1832  ;
;1833  ; START OF REVISION HISTORY:
;1834  ;
;1835  ;
;1836  ;
;1837  ;
;1838  ;
;1839  ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D, 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.100A  ;1874 TRPH0: Q_ID[USTACK]
U 100A, 0C00.003C.01E0.0800.0000.100E  ;1875  Q_D,D_Q
U 100E, 0000.003C.8180.3C00.0000.101A  ;1876  ID[USTACK]_D
U 101A, 0C00.003C.01E0.0800.0000.101E  ;1877  Q_D,D_Q
U 101E, 0000.003C.01C0.0A78.0000.104F  ;1878  Q_R[0F]
U 104F, 0001.2024.0180.0800.0010.105B  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 105B, 0000.013C.0180.0800.0000.1002  ;1880  Z?    ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004, 0000.003D.D980.0800.0084.719A  ;1898 =0 SETRCF[.9]
U 1005, 0000.003C.49F0.2C00.0000.105F  ;1899  Q_ID[TBER0]
U 105F, 0001.203C.4DF0.2DB0.0000.1109  ;1900  RC[6]_Q,Q_ID[TBER1]
U 1109, 0001.203C.65F0.2DB8.0000.1154  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 1154, 0001.203C.6DF0.2DD8.0000.117D  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 117D, 0001.203C.75F0.2DE0.0000.117E  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 117E, 0001.203C.79F0.2DC8.0000.117F  ;1904  RC[9]_Q,Q_ID[PARITY]
U 117F, 0001.203C.69F0.2DC0.0000.1180  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 1180, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.1194  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 1181, 0000.003C.8580.0800.0084.7182  ;1916 SCOPE: SC_K[.C]
U 1182, 0803.001C.0180.0800.0000.1006  ;1917  ALU_NOT.MASK,D_ALU
U 1006, 0000.003D.7580.3C00.0000.11DD  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1007, 0000.003C.65F8.0800.0084.7183  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 1183, 0818.0038.8D80.0800.0000.1184  ;1920  D_K[.1F]    ; ...
U 1184, 0D00.003C.0180.0800.0000.1185  ;1921  D_DAL.SC
U 1185, 0000.003C.3D80.3C00.0000.1186  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 1186, 0818.0038.0580.0800.0000.1187  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 1187, 0D00.003C.0180.0800.0000.1188  ;1924  D_DAL.SC
U 1188, 0F00.003C.3180.3C00.0000.1189  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1189, 0000.003C.0180.0800.0000.118A  ;1926  NOP
U 118A, 0000.003C.3980.3C00.0000.118B  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 118B, 0000.003C.2980.3C00.0000.118C  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 118C, 0000.003C.4980.3C00.0000.118D  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 118D, 0818.0038.C180.0800.0000.118E  ;1930  D_K[.FFFF]
U 118E, 0000.003C.6580.3C00.0000.118F  ;1931  ID[SBI.ERR]_D
U 118F, 0F00.003C.6D80.3C00.0000.1190  ;1932  ID[FAULT]_D,D_0
U 1190, 0000.003C.6D80.3C00.0000.1191  ;1933  ID[FAULT]_D
U 1191, 0000.003C.6580.3C00.0000.1192  ;1934  ID[SBI.ERR]_D
U 1192, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 1193, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 1194, 0810.0038.0180.0978.0000.1195  ;1937 ERROR1: D_RC[0F]
U 1195, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 1196, 0810.0038.0180.0978.0000.1197  ;1940 ERROR2: D_RC[0F]
U 1197, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155, 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 1198, 0810.0038.4D80.0978.0000.1199  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1199, 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 119A, 0010.0038.01C0.0978.0000.119B  ;1980 SETRCF: Q_RC[0F]
U 119B, 0019.2034.4DC0.0800.0000.119C  ;1981  Q_Q.AND.K[.FF00]
U 119C, 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983
            ;1984 ;*******************************************************
            ;1985 ;
            ;1986 ; MAIN MACHINE/FPA SUBROUTINES
            ;1987 ;
            ;1988 ;*******************************************************
            ;1989
            ;1990 ; SHIFT D REGISTTER CONTENTS
            ;1991
            ;1992
            ;1993 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
            ;1994
            ;1995
U 119D, 0000.003C.01F8.0800.0000.119E  ;1996 SHIFTD: Q_0
U 119E, 0500.003C.0180.0800.0000.119F  ;1997  D_D.LEFT  ;
U 119F, 0000.003C.0580.0800.1414.B1A0  ;1998  EALU_STATE-K[.1],CLK.UBCC ;
U 11A0, 0000.123C.0180.0800.0000.100B  ;1999  EALU.Z?
U 100B, 0000.003C.0180.0800.0000.119D  ;2000 =1011 J/SHIFTD  ;
U 100F, 0000.003E.0180.0800.0000.0001  ;2001  RETURN1
            ;2002
            ;2003
            ;2004
            ;2005
```

```
            ;2006
            ;2007 ;*
            ;2008 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
            ;2009 ; OR'S IN BIT 15, SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP TO
            ;2010 ; THAT ADDRESS.
            ;2011 ;-
            ;2012
U 11A1, 0819.0030,4580,0800,0000,11A2  ;2013 TRAP1: D_D.OR.K[.8000]
U 11A2, 0000.003C,6580,0800,1404,7008  ;2014  STATE_K[.10]
U 1008, 0000.003D,0180,0800,0000,119D  ;2015 =0 CALL[SHIFTD]
U 1009, 0000.003C,5980,3C00,0000,11A3  ;2016  ID[ACC.2]_D
U 11A3, 0000.00FE,0380,0800,0000,0001  ;2017  TRAP.FPA,RETURN1
            ;2018
            ;2019
            ;2020 ;*
            ;2021 ; THIS ROUTINE USES THE CONTENTS OF THE PREVIOUSLY LOADED STATE
            ;2022 ; REGISTER AS A COUNTER.
            ;2023 ;-
            ;2024
U 11A4, 0000.007C,0180,0800,0000,11A5  ;2025 STEP: CPSYNC  ;
U 11A5, 0000.003C,0580,0800,1414,B1A6  ;2026  EALU_STATE-K[.1],CLK.UBCC
U 11A6, 0000.123C,0180,0800,0000,101B  ;2027  EALU.Z?
U 101B, 0000.003C,0180,0800,0000,11A4  ;2028 =1011 J/STEP
U 101F, 0000.003E,0180,0800,0000,0001  ;2029  RETURN1
            ;2030
            ;2031
            ;2032
            ;2033
            ;2034
            ;2035 ;*
            ;2036 ; THE FOLLOWING ROUTINES GENERATE FPA UPC ADDRESSES, AND
            ;2037 ; JUMP TO THE TRAP ROUTINE.
            ;2038 ;-
            ;2039
U 11A7, 0818.0038,4180,0800,0000,11A8  ;2040 TRP.85: D_K[.80]
U 11A8, 0819.0010,1180,0800,0000,11A1  ;2041  D_D+K[.4]+1,J/TRAP1
            ;2042
            ;2043
U 11A9, 0818.0038,4180,0800,0000,11AA  ;2044 TRP.87: D_K[.80]
U 11AA, 0000.003C,5D80,0800,0000,11AB  ;2045  KMX/.7
U 11AB, 0819.0014,5D80,0800,0000,11A1  ;2046  D_D+K[.7],J/TRAP1
            ;2047
            ;2048
            ;2049
            ;2050
            ;2051
U 11AC, 0818.0038,CD80,0800,0000,11AD  ;2052 TRP.F1: D_K[.F0]
U 11AD, 0819.0014,0580,0800,0000,11A1  ;2053  D_D+K[.1],J/TRAP1
            ;2054
            ;2055
U 11AE, 0818.0038,CD80,0800,0000,11AF  ;2056 TRP.F2: D_K[.F0]
U 11AF, 0819.0014,0980,0800,0000,11A1  ;2057  D_D+K[2],J/TRAP1
            ;2058
            ;2059
U 11B0, 0818.0038,CD80,0800,0000,11B1  ;2060 TRP.F3: D_K[.F0]
```

```
U 11B1, 0819,0014,0D80,0800,0000,11A1  ;2061  D_D+K[.3],J/TRAP1
            ;2062
            ;2063
            ;2064
U 11B2, 0818,0038,CD80,0800,0000,11B3  ;2065 TRP.F7: D_K[.F0]
U 11B3, 0000,003C,5D80,0800,0000,11B4  ;2066  KMX/.7
U 11B4, 0819,0014,5D80,0800,0000,11A1  ;2067  D_D+K[.7],J/TRAP1
            ;2068
            ;2069
U 11B5, 0818,0038,CD80,0800,0000,11B6  ;2070 TRP.F8: D_K[.F0]
U 11B6, 0819,0014,0180,0800,0000,11A1  ;2071  D_D+K[.8],J/TRAP1
            ;2072
            ;2073 ;*
            ;2074 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2075 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE IB
            ;2076 ; IS COMPLETELY FULL OF DATA.
            ;2077 ;
            ;2078 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2079 ;-
U 11B7, 3000,003C,0180,6000,0000,11B8  ;2080 IBLOAD: LOAD.IB,IBC/START
U 11B8, 0000,003C,0180,F800,0000,11B9  ;2081 SYNC02: MCT/ALLOW.IB.READ
U 11B9, 0000,003C,0180,F800,0000,11BA  ;2082  MCT/ALLOW.IB.READ
U 11BA, 0000,003C,0180,F800,0000,11BB  ;2083  MCT/ALLOW.IB.READ
U 11BB, 0000,003C,0180,F800,0000,11BC  ;2084  MCT/ALLOW.IB.READ
U 11BC, 0000,003C,0180,F800,0000,11BD  ;2085  MCT/ALLOW.IB.READ
U 11BD, 1000,003E,0180,F800,0000,0001  ;2086  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2087
            ;2088 ;*
            ;2089 ; THESE ROUTINES WILL LOAD THE EXPONENT AND SIGN DATA INTO THE FPA
            ;2090
            ;2091 ; LA>LB,SA=0,SB=0
            ;2092
U 11BE, 0818,0038,2980,0800,0000,100C  ;2093 LDDATA: D_K[.34]
U 100C, 0000,003D,5D80,0800,1404,719D  ;2094 =0 STATE_K[.7],CALL[SHIFTD]
U 100D, 0001,003C,0180,0AA8,0000,11BF  ;2095  R[5]_D
U 11BF, 0818,0038,2D80,0800,0000,1010  ;2096  D_K[.28]
U 1010, 0000,003D,5D80,0800,1404,719D  ;2097 =0 STATE_K[.7],CALL[SHIFTD]
U 1011, 0001,003E,0180,0AB0,0000,0001  ;2098  R[6]_D,RETURN1
            ;2099
            ;2100 ; LA>LB,SA=0,SB=1
            ;2101
U 11C0, 0818,0038,2980,0800,0000,1012  ;2102 LDDAT1: D_K[.34]
U 1012, 0000,003D,5D80,0800,1404,719D  ;2103 =0 STATE_K[.7],CALL[SHIFTD]
U 1013, 0001,003C,0180,0AA8,0000,11C1  ;2104  R[5]_D
U 11C1, 0818,0038,2D80,0800,0000,1014  ;2105  D_K[.28]
U 1014, 0000,003D,5D80,0800,1404,719D  ;2106 =0 STATE_K[.7],CALL[SHIFTD]
U 1015, 0819,0030,4580,0800,0000,11C2  ;2107  D_D.OR.K[.8000]  ; SB=1
U 11C2, 0001,003E,0180,0AB0,0000,0001  ;2108  R[6]_D,RETURN1
            ;2109
            ;2110 ; LA<LB,SA=0,SB=0
            ;2111
U 11C3, 0818,0038,2980,0800,0000,1016  ;2112 LDDAT2: D_K[.34]
U 1016, 0000,003D,5D80,0800,1404,719D  ;2113 =0 STATE_K[.7],CALL[SHIFTD]
U 1017, 0001,003C,0180,0AB0,0000,11C4  ;2114  R[6]_D
U 11C4, 0818,0038,0180,0800,0000,1018  ;2115  D_K[.8]
```

```
U 1018. 0000.003D.5D80.0800.1404.719D  ;2116 =0 STATE_K[.7],CALL[SHIFTD]
U 1019. 0001.003E.0180.0AA8.0000.0001  ;2117  R[5]_D,RETURN1
            ;2118
            ;2119
            ;2120 ; LA<LB,SA=0,SB=1
            ;2121
U 11C5. 0818.0038.2D80.0800.0000.101C  ;2122 LDDAT3: D_K[.28]
U 101C. 0000.003D.5D80.0800.1404.719D  ;2123 =0 STATE_K[.7],CALL[SHIFTD]
U 101D. 0001.003C.0180.0AA8.0000.11C6  ;2124  R[5]_D
U 11C6. 0818.0038.2980.0800.0000.1020  ;2125  D_K[.34]
U 1020. 0000.003D.5D80.0800.1404.719D  ;2126 =0 STATE_K[.7],CALL[SHIFTD]
U 1021. 0819.0030.4580.0800.0000.11C7  ;2127  D_D.OR.K[.8000]  ; SB=1
U 11C7. 0001.003E.0180.0AB0.0000.0001  ;2128  R[6]_D,RETURN1
            ;2129
            ;2130
            ;2131 ; LA>LB,SA=1,SB=1
            ;2132
U 11C8. 0818.0038.2980.0800.0000.1022  ;2133 LDDAT4: D_K[.34]
U 1022. 0000.003D.5D80.0800.1404.719D  ;2134 =0 STATE_K[.7],CALL[SHIFTD]
U 1023. 0819.0030.4580.0800.0000.11C9  ;2135  D_D.OR.K[.8000]  ; SA=1
U 11C9. 0001.003C.0180.0AA8.0000.11CA  ;2136  R[5]_D
U 11CA. 0818.0038.2D80.0800.0000.1024  ;2137  D_K[.28]
U 1024. 0000.003D.5D80.0800.1404.719D  ;2138 =0 STATE_K[.7],CALL[SHIFTD]
U 1025. 0819.0030.4580.0800.0000.11CB  ;2139  D_D.OR.K[.8000]  ; SB=1
U 11CB. 0001.003E.0180.0AB0.0000.0001  ;2140  R[6]_D,RETURN1
            ;2141
            ;2142 ; LA=LB,SA=1,SB=0
            ;2143
U 11CC. 0818.003B.2980.0800.0000.1026  ;2144 LDDAT5: D_K[.34]
U 1026. 0000.003D.5D80.0800.1404.719D  ;2145 =0 STATE_K[.7],CALL[SHIFTD]
U 1027. 0001.003C.0180.0AB0.0000.11CD  ;2146  R[6]_D
U 11CD. 0819.0030.4580.0800.0000.11CE  ;2147  D_D.OR.K[.8000]  ; SA=1
U 11CE. 0001.003E.0180.0AA8.0000.0001  ;2148  R[5]_D,RETURN1
            ;2149
            ;2150 ; LA<LB,SA=1,SB=1
            ;2151
U 11CF. 0818.0038.D180.0800.0000.11D0  ;2152 LDDAT6: D_K[.C0]
U 11D0. 0819.0014.0D80.0800.0000.1028  ;2153  D_D+K[.3]
U 1028. 0000.003D.5D80.0800.1404.719D  ;2154 =0 STATE_K[.7],CALL[SHIFTD]
U 1029. 0819.0030.4580.0800.0000.11D1  ;2155  D_D.OR.K[.8000]
U 11D1. 0001.003C.0180.0AB0.0000.11D2  ;2156  R[6]_D
U 11D2. 0818.0038.D180.0800.0000.102A  ;2157  D_K[.C0]
U 102A. 0000.003D.5D80.0800.1404.719D  ;2158 =0 STATE_K[.7],CALL[SHIFTD]
U 102B. 0819.0030.4580.0800.0000.11D3  ;2159  D_D.OR.K[.8000]
U 11D3. 0001.003E.0180.0AA8.0000.0001  ;2160  R[5]_D,RETURN1
            ;2161
            ;2162
            ;2163
            ;2164
            ;2165
            ;2166 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRSSS IN THE D REGISTER AND
            ;2167 ; CONVERT IT TO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2168 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2169
U 11D4. 0000.003C.65F8.0800.0084.71D5  ;2170 GENTRA: SC_K[.10],Q_0
```

```
U 11D5, 0D00,003C,8D80,0800,0084,71D6  ;2171  D_DAL.SC,SC_K[.1F]
U 11D6, 0801,001E,0180,0800,0000,0001  ;2172  ALU_D.ORNOT.MASK,D_ALU,RETURN1
          ;2173
          ;2174
          ;2175 ;*
          ;2176 ; THIS ROUTINE LOADS THE PR IN THE EXPONENT PROCESSOR, FROM RS.
          ;2177 ; THE PR IS ROUTED TO THE EALU OUTPUT VIA THE AMX. THE PR PROVIDES
          ;2178 ; A NON-ZERO OUTPUT WHICH PREVENTS THE NORMALIZER MUX FORCING ZEROS
          ;2179 ; ON THE BUS, WHEN THE SIGN PROCESSOR DATA IS READ BACK TO THE CPU.
          ;2180
U 11D7, 0818,0038,E980,0800,0000,11D8  ;2181 LD.PR: D_K[.24]
U 11D8, 0000,003C,0180,0800,0000,102C  ;2182  NOP
U 102C, 0819,0015,0580,0800,0000,11D4  ;2183 =0 D_D*K[.1],CALL[GENTRA]
U 102D, 0000,003C,5980,3C00,0000,11D9  ;2184  ID[ACC.2]_D
U 11D9, 0000,00FC,0380,0A28,0000,11DA  ;2185  TRAP.FPA,LAB_R[5]
U 11DA, 0000,007C,0180,0A28,0000,11DB  ;2186  LAB_R[5],CPSYNC ; FPA AT 025. LOAD LA
U 11DB, 0000,003C,0180,0800,0000,11DC  ;2187  NOP    ; FPA AT 026. LOAD PR
U 11DC, 0000,003E,0180,0800,0000,0001  ;2188  RETURN1
          ;2189
          ;2190
          ;2191
          ;2192 ;
          ;2193 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
          ;2194 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
          ;2195 ;   28: NAD/28  ; NOP
          ;2196 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
          ;2197 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
          ;2198 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
          ;2199 ;  INITIALIZE ITSELF.
          ;2200 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
          ;2201 ;
U 11DD, 0818,0038,4580,0800,0000,11DE  ;2202 FPINIT: D_K[.8000]
U 11DE, 0000,003C,5D80,3C00,0000,102E  ;2203  ID[ACC.CS]_D  ; TURN ON FPA
U 102E, 0818,0039,2D80,0800,0000,11D4  ;2204 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 102F, 0000,003C,5980,3C00,0000,11DF  ;2205  ID[ACC.2]_D
U 11DF, 0000,00FC,0380,0800,0000,1030  ;2206  TRAP.FPA  ; TRAP FPA TO 28.
U 1030, 0018,0039,1980,0800,0200,11E5  ;2207 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 1031, 0F00,003C,0180,0800,0000,1032  ;2208  D_0
U 1032, 0000,003D,0180,0800,0000,11D4  ;2209 =0 CALL,J/GENTRA
U 1033, 0000,003C,5980,3C00,0000,11E0  ;2210  ID[ACC.2]_D
U 11E0, 0000,00FC,0380,0800,0000,1034  ;2211  TRAP.FPA  ; TRAP FPA TO 0
U 1034, 0818,0039,2D80,0800,0000,11D4  ;2212 =0 D_K[.28],CALL,J/GENTRA
U 1035, 0000,003C,5980,3C00,0000,11E1  ;2213  ID[ACC.2]_D
U 11E1, 0000,00FE,0380,0800,0000,0001  ;2214  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
          ;2215 ;
          ;2216 ;
          ;2217 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
          ;2218 ;%0
          ;2219 ;$ 0000,0000, 0000,0000
          ;2220 ;
          ;2221 ;
          ;2222 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
          ;2223 ; BY TRAPPING THE FPA TO LOCATION 0.
          ;2224 ;
          ;2225 =0
```

```
U 1036. 0F00.003D.0180.0800.0000.11D4  ;2226 FPIRD: D_0,CALL,J/GENTRA
U 1037. 0000.003C.5980.3C00.0000.11E2  ;2227  ID[ACC.2]_D
U 11E2. 0000.00FC.0380.0800.0000.11E3  ;2228  TRAP.FPA  ; TRAP TO FPA 0
U 11E3. 0000.003C.0180.0800.0000.11E4  ;2229  NOP
U 11E4. 0000.003E.0180.0800.0000.0001  ;2230  RETURN1
          ;2231 ;
          ;2232 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
          ;2233 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
          ;2234 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
          ;2235 ;
U 11E5. 2000.003C.0180.0800.0000.11E6  ;2236 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 11E6. 3000.003C.0180.6000.0000.11E7  ;2237  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 11E7. 0000.003C.0180.F800.0000.11E8  ;2238 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 11E8. 0000.003C.0180.F800.0000.11E9  ;2239  MCT/ALLOW.IB.READ
U 11E9. 0000.003C.0180.F800.0000.11EA  ;2240  MCT/ALLOW.IB.READ
U 11EA. 0000.003C.0180.F800.0000.11EB  ;2241  MCT/ALLOW.IB.READ
U 11EB. 0000.003C.0180.F800.0000.11EC  ;2242  MCT/ALLOW.IB.READ
U 11EC. 1000.003C.0180.F800.0000.11ED  ;2243  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 11ED. 0000.003E.0180.0800.0000.0001  ;2244  RETURN1
U 11EE. 3000.003C.0180.0800.0000.11E7  ;2245 WAITIB: IBC/START,J/IBW
          ;2246
          ;2247 ; THESE ROUTINES ARE RESERVED FOR THE WCS DEBUGGER.
          ;2248
          ;2249
          ;2250
U 1120. 0000.003C.0180.0800.0000.1121  ;2251 1120: NOP
U 1121. 0000.003C.0180.0800.0000.1122  ;2252 1121: NOP
U 1122. 0000.003C.0180.0800.0000.1123  ;2253 1122: NOP
U 1123. 0000.003C.0180.0800.0000.1124  ;2254 1123: NOP
U 1124. 0000.003C.0180.0800.0000.1125  ;2255 1124: NOP
U 1125. 0000.003C.0180.0800.0000.1126  ;2256 1125: NOP
U 1126. 0000.003C.0180.0800.0000.1127  ;2257 1126: NOP
U 1127. 0000.003C.0180.0800.0000.1128  ;2258 1127: NOP
U 1128. 0000.003C.0180.0800.0000.1129  ;2259 1128: NOP
U 1129. 0000.003C.0180.0800.0000.112A  ;2260 1129: NOP
U 112A. 0000.003C.0180.0800.0000.112B  ;2261 112A: NOP
U 112B. 0000.003C.0180.0800.0000.112C  ;2262 112B: NOP
U 112C. 0000.003C.0180.0800.0000.112D  ;2263 112C: NOP
U 112D. 0000.003C.0180.0800.0000.112E  ;2264 112D: NOP
U 112E. 0000.003C.0180.0800.0000.112F  ;2265 112E: NOP
U 112F. 0000.003C.0180.0800.0000.1038  ;2266 112F: NOP
```

```
                ;2267   .PAGE "TEST 27D SIGN LOGIC FOR MUL/DIV"
                ;2268   ;*****************************************************************
                ;2269   ;**
                ;2270   ; TEST SIGN LOGIC FOR MUL/DIV
                ;2271   ;
                ;2272   ; TEST DESCRIPTION
                ;2273   ;
                ;2274   ; THE SIGN DECISION LOGIC FOR MULTIPLY AND DIVIDE IS
                ;2275   ; EXERCISED IN THIS TEST. THE RESULTING SIGN OUTPUT IS CHECKED
                ;2276   ; WITH ALL INPUTS ASSERTED. THEN, EACH INPUT IS DEASSERTED,
                ;2277   ; IN TURN, AND THE RESULTING SIGN OUTPUT IS VERIFIED.
                ;2278   ; FPA UCODE USED
                ;2279   ; LOCATION CONTENTS
                ;2280   ; 0F3: BSC/NH,SGNC/A.RES,NAD/PSTALL
                ;2281   ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
                ;2282   ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,AMXC/LA,NAD/0F6
                ;2283   ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
                ;2284   ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/027
                ;2285   ; 027: BSC/R,SCR/CPU,EAC/LDAB,WAIT,NAD/02B
                ;2286   ;
                ;2287   ; LOGIC DESCRIPTION
                ;2288   ;
                ;2289   ; ALL OF THE SIGN DECISION LOGIC FOR MULTIPLY
                ;2290   ; AND DIVIDE IS LOCATED ON THE FCT MODULE.
                ;2291   ;
                ;2292   ; ERROR DESCRIPTION
                ;2293   ; EXPECTED SIGN RESULT
                ;2294   ; RECEIVED SIGN RESULT
                ;2295   ;
                ;2296   ; SYNC POINT DESCRIPTION
                ;2297   ;
                ;2298   ;--
                ;2299   ;*****************************************************************
                ;2300   ;/////////////////////////////////////////
                ;2301   ;*
                ;2302   ; SUBTEST 1
                ;2303   ; (SA(0)H.XOR.-SB(0)H)*(MUL/DIVH)*(-POLYADDL)
                ;2304   ; TEST FOR SGNC/A.RES
                ;2305   ;-
                ;2306   =0
U 1038, 001B.0039,0980,09F8,0000,1181   ;2307 SGN2: NEWTST[.2]
U 1039, 001B.0038,4580,09F0,0000,103A   ;2308  RC[0E]_K[.8000]   ; EXPECTED SIGN RESULT
U 103A, 0000.003D,0180,0800,0000,1198   ;2309 =0 SUBTEST
U 103B, 0000.003C,0180,0800,0000,103C   ;2310  NOP
U 103C, 0000.003D,0180,0800,0000,11C0   ;2311 =0 CALL[LDDAT1]   ; LA>LB,SA=0,SB=1
U 103D, 0000.003C,0180,0800,0000,103E   ;2312  NOP
U 103E, 0000.003D,0180,0800,0000,11D7   ;2313 =0 CALL[LD.PR]   ; SET EALU <7:0> NOT EQUAL TO ZERO
U 103F, 0000.003C,0180,0800,0000,1040   ;2314  NOP
U 1040, 201B.0039,2180,0800,4200,11B7   ;2315 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD]   ; OPCODE IS MULF R,R
U 1041, 0000.003C,0180,0800,0000,1042   ;2316  NOP
U 1042, 0000.003D,0180,0800,0000,11B5   ;2317 =0 CALL[TRP.F8]   ; LOAD LA,LB,SA,SB
U 1043, 0000.007C,0180,0800,0000,11EF   ;2318  CPSYNC
U 11EF, 0000.003C,01D8,0800,0000,11F0   ;2319  Q_ACC   ;
U 11F0, 0019.2034,45C0,0800,0000,11F1   ;2320  Q_Q.AND.K[.8000]
U 11F1, 0001.203C,0180,09E8,0000,11F2   ;2321  RC[0D]_Q
```

```
U 11F2. 0011.2020.0180.0970.0010.11F3  ;2322  ALU_Q.XOR.RC[0E],CLK.UBCC
U 11F3. 0000.013C.0180.0800.0000.1044  ;2323  Z?
U 1044. 0000.003D.0180.0800.0000.1194  ;2324 =0 ERROR1
U 1045. 0000.003C.0180.0800.0000.1046  ;2325  NOP
          ;2326 ;/////////////////////////////////////////
          ;2327 ;*
          ;2328 ; SUBTEST 2
          ;2329 ; (-SA(0)H.XOR.SB(0)H)*(MUL/DIVH)*(-POLYADDL)
          ;2330 ;-
U 1046. 0000.003D.0180.0800.0000.1198  ;2331 =0 SUBTEST
U 1047. 0000.003C.0180.0800.0000.1048  ;2332  NOP
U 1048. 0000.003D.0180.0800.0000.1193  ;2333 =0 ERLOOP
U 1049. 0000.003C.0180.0800.0000.11F4  ;2334  NOP
U 11F4. 0018.0038.4580.09F0.0000.104A  ;2335  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 104A. 0000.003D.0180.0800.0000.11BE  ;2336 =0 CALL[LDDATA]  ; LA>LB,SA=0,SB=0
U 104B. 0800.003C.0180.0A28.0000.11F5  ;2337  D_R[5]
U 11F5. 0019.0030.4580.0AA8.0000.104C  ;2338  R[5]_D.OR.K[.8000] ; SA=1
U 104C. 0000.003D.0180.0800.0000.11D7  ;2339 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 104D. 0000.003C.0180.0800.0000.1050  ;2340  NOP
U 1050. 2018.0039.6580.0800.4200.11B7  ;2341 =0 ALU_K[.10],FLUSH.IB,CALL[IBLOAD]  ; OPCODE IS DIVF R,R
U 1051. 0000.003C.0180.0800.0000.1052  ;2342  NOP
U 1052. 0000.003D.0180.0800.0000.11B5  ;2343 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1053. 0000.007C.0180.0800.0000.11F6  ;2344  CPSYNC
U 11F6. 0000.003C.01D8.0800.0000.11F7  ;2345  Q_ACC  ;
U 11F7. 0019.2034.45C0.0800.0000.11F8  ;2346  Q_Q.AND.K[.8000]
U 11F8. 0001.203C.0180.09E8.0000.11F9  ;2347  RC[0D]_Q
U 11F9. 0011.2020.0180.0970.0010.11FA  ;2348  ALU_Q.XOR.RC[0E],CLK.UBCC
U 11FA. 0000.013C.0180.0800.0000.1054  ;2349  Z?
U 1054. 0000.003D.0180.0800.0000.1194  ;2350 =0 ERROR1
U 1055. 0000.003C.0180.0800.0000.1056  ;2351  NOP
          ;2352 ;/////////////////////////////////////////
          ;2353 ;*
          ;2354 ; SUBTEST 3
          ;2355 ; (SA(0)H.XOR.SB(0)H)*(MUL/DIVH)*(-POLYADDL)
          ;2356 ;-
U 1056. 0000.003D.0180.0800.0000.1198  ;2357 =0 SUBTEST
U 1057. 0000.003C.0180.0800.0000.1058  ;2358  NOP
U 1058. 0000.003D.0180.0800.0000.1193  ;2359 =0 ERLOOP
U 1059. 0000.003C.0180.0800.0000.11FB  ;2360  NOP
U 11FB. 0003.003C.0180.09F0.0000.105C  ;2361  RC[0E]_0  ; EXPECTED SIGN RESULT
U 105C. 0000.003D.0180.0800.0000.11BE  ;2362 =0 CALL[LDDATA]  ; LA>LB,SA=0,SB=0
U 105D. 0000.003C.0180.0800.0000.1060  ;2363  NOP
U 1060. 0000.003D.0180.0800.0000.11D7  ;2364 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1061. 0000.003C.0180.0800.0000.1062  ;2365  NOP
U 1062. 2018.0039.2180.0800.4200.11B7  ;2366 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD]  ; OPCODE IS MULF R,R
U 1063. 0000.003C.0180.0800.0000.1064  ;2367  NOP
U 1064. 0000.003D.0180.0800.0000.11B5  ;2368 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1065. 0000.007C.0180.0800.0000.11FC  ;2369  CPSYNC
U 11FC. 0000.003C.01D8.0800.0000.11FD  ;2370  Q_ACC  ;
U 11FD. 0019.2034.45C0.0800.0000.11FE  ;2371  Q_Q.AND.K[.8000]
U 11FE. 0001.203C.0180.09E8.0000.11FF  ;2372  RC[0D]_Q
U 11FF. 0011.2020.0180.0970.0010.1200  ;2373  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1200. 0000.013C.0180.0800.0000.1066  ;2374  Z?
U 1066. 0000.003D.0180.0800.0000.1194  ;2375 =0 ERROR1
U 1067. 0000.003C.0180.0800.0000.1068  ;2376  NOP
```

```
            ;2377 ;//////////////////////////////////////////
            ;2378 ;*
            ;2379 ; SUBTEST 4
            ;2380 ; (-SA(0)H.XOR.-SB(0)H)*(MUL/DIV)*(-POLYADDL)
            ;2381 ;-
U 1068. 0000.003D.0180.0800.0000.1198  ;2382 =0 SUBTEST
U 1069. 0000.003C.0180.0800.0000.106A  ;2383  NOP
U 106A. 0000.003D.0180.0800.0000.1193  ;2384 =0 ERLOOP
U 106B. 0000.003C.0180.0800.0000.1201  ;2385  NOP
U 1201. 0003.003C.0180.09F0.0000.106C  ;2386  RC[0E]_0   ; EXPECTED SIGN RESULT
U 106C. 0000.003D.0180.0800.0000.11C8  ;2387 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 106D. 0000.003C.0180.0800.0000.106E  ;2388  NOP
U 106E. 0000.003D.0180.0800.0000.11D7  ;2389 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 106F. 0000.003C.0180.0800.0000.1070  ;2390  NOP
U 1070. 2018.0039.6580.0800.4200.11B7  ;2391 =0 ALU_K[.10],FLUSH.IB,CALL[IBLOAD]  ; OPCODE IS DIVF R,R
U 1071. 0000.003C.0180.0800.0000.1072  ;2392  NOP
U 1072. 0000.003D.0180.0800.0000.11B5  ;2393 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1073. 0000.007C.0180.0800.0000.1202  ;2394  CPSYNC
U 1202. 0000.003C.01D8.0800.0000.1203  ;2395  Q_ACC  ;
U 1203. 0019.2034.45C0.0800.0000.1204  ;2396  Q_Q.AND.K[.8000]
U 1204. 0001.203C.0180.09E8.0000.1205  ;2397  RC[0D]_Q
U 1205. 0011.2020.0180.0970.0010.1206  ;2398  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1206. 0000.013C.0180.0800.0000.1074  ;2399  Z?
U 1074. 0000.003D.0180.0800.0000.1194  ;2400 =0 ERROR1
U 1075. 0000.003C.0180.0800.0000.1076  ;2401  NOP
            ;2402 ;//////////////////////////////////////////
            ;2403 ;*
            ;2404 ; SUBTEST 5
            ;2405 ; (SA(0)H.XOR.-SB(0)H)*(-MUL/DIVH)*(-POLYADDL)
            ;2406 ;-
U 1076. 0000.003D.0180.0800.0000.1198  ;2407 =0 SUBTEST
U 1077. 0000.003C.0180.0800.0000.1078  ;2408  NOP
U 1078. 0000.003D.0180.0800.0000.1193  ;2409 =0 ERLOOP
U 1079. 0000.003C.0180.0800.0000.1207  ;2410  NOP
U 1207. 0003.003C.0180.09F0.0000.107A  ;2411  RC[0E]_0   ; EXPECTED SIGN RESULT
U 107A. 0000.003D.0180.0800.0000.11C0  ;2412 =0 CALL[LDDAT1]  ; LA>LB,SA=0,SB=1
U 107B. 0000.003C.0180.0800.0000.107C  ;2413  NOP
U 107C. 0000.003D.0180.0800.0000.11D7  ;2414 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 107D. 0000.003C.0180.0800.0000.107E  ;2415  NOP
U 107E. 2018.0039.0180.0800.4200.11B7  ;2416 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD]  ; OPCODE IS ADDF R,R
U 107F. 0000.003C.0180.0800.0000.1080  ;2417  NOP
U 1080. 0000.003D.0180.0800.0000.11B5  ;2418 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1081. 0000.007C.0180.0800.0000.1208  ;2419  CPSYNC
U 1208. 0000.003C.01D8.0800.0000.1209  ;2420  Q_ACC  ;
U 1209. 0019.2034.45C0.0800.0000.120A  ;2421  Q_Q.AND.K[.8000]
U 120A. 0001.203C.0180.09E8.0000.120B  ;2422  RC[0D]_Q
U 120B. 0011.2020.0180.0970.0010.120C  ;2423  ALU_Q.XOR.RC[0E],CLK.UBCC
U 120C. 0000.013C.0180.0800.0000.1082  ;2424  Z?
U 1082. 0000.003D.0180.0800.0000.1194  ;2425 =0 ERROR1
U 1083. 0000.003C.0180.0800.0000.1084  ;2426  NOP
            ;2427 ;//////////////////////////////////////////
            ;2428 ;*
            ;2429 ; SUBTEST 6
            ;2430 ; (SA(0)H.XOR.-SB(0)H)*(MUL/DIV)*(POLYADDL)
            ;2431 ;-
```

```
ZZ-ESKAR-V22  TEST 27D SIGN LOGIC FOR MUL/DIV                                   SEQ 2641
ESKAR61.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:17           Page   6

U 1084. 0000.003D.0180.0800.0000.1198  ;2432 =0 SUBTEST
U 1085. 0000.003C.0180.0800.0000.1086  ;2433  NOP
U 1086. 0000.003D.0180.0800.0000.1193  ;2434 =0 ERLOOP
U 1087. 0000.003C.0180.0800.0000.120D  ;2435  NOP
U 120D. 0003.003C.0180.09F0.0000.1088  ;2436  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1088. 0000.003D.0180.0800.0000.11C0  ;2437 =0 CALL[LDDAT1]  ; LA>LB,SA=0,SB=1
U 1089. 0000.003C.0180.0800.0000.108A  ;2438  NOP
U 108A. 0000.003D.0180.0800.0000.11D7  ;2439 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 108B. 0000.003C.0180.0800.0000.108C  ;2440  NOP
U 108C. 2018.0039.2180.0800.4200.11B7  ;2441 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD]  ; OPCODE IS MULF R,R
U 108D. 0000.003C.0180.0800.0000.108E  ;2442  NOP
U 108E. 0000.003D.0180.0800.0000.11B5  ;2443 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 108F. 0000.003C.0180.0800.0000.1090  ;2444  NOP  ; SA_BUSA(15),SB_BUSB(15),
        ;2445    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 1090. 0000.003D.0180.0800.0000.11B0  ;2446 =0 CALL[TRP.F3]  ; ASSERT POLYADDL
U 1091. 0000.003C.0180.0800.0000.120E  ;2447  NOP
U 120E. 0000.003C.01D8.0800.0000.120F  ;2448  Q_ACC
U 120F. 0019.2034.45C0.0800.0000.1210  ;2449  Q_Q.AND.K[.8000]
U 1210. 0001.203C.0180.09E8.0000.1211  ;2450  RC[0D]_Q
U 1211. 0011.2020.0180.0970.0010.1212  ;2451  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1212. 0000.013C.0180.0800.0000.1092  ;2452  Z?
U 1092. 0000.003D.0180.0800.0000.1194  ;2453 =0 ERROR1
U 1093. 0000.003C.0180.0800.0000.1094  ;2454  NOP
```

```
;2455  .PAGE "TEST 27E SIGN LOGIC FOR EXACT ZERO, UFLO, OVFLO"
;2456 ;*****************************************************************
;2457 ;**
;2458 ; TEST SIGN LOGIC FOR EXACT ZERO, UFLO, OVFLO
;2459 ;
;2460 ; TEST DESCRIPTION
;2461 ;
;2462 ; THE LOGIC WHICH CONTROLS THE SIGN DECISION FOR EXACT
;2463 ; ZERO, OVERFLOW, AND UNDERFLOW, IS EXERCISED IN THIS
;2464 ; TEST. THE SIGN OUTPUT IS CHECKED WITH ALL INPUTS ASSERTED.
;2465 ; THEN, EACH INPUT IS DEASSERTED, IN TURN, AND THE
;2466 ; RESULTING SIGN OUTPUT IS VERIFIED.
;2467 ;
;2468 ;
;2469 ; FPA UCODE USED
;2470 ; LOCATION CONTENTS
;2471 ; 0F0: EAC/LDPR,AMXC/LA,EALUC/A,SGNC/NOP,WAIT,NAD/0EF
;2472 ; 0EF: BSC/NH,SGNC/A.RES,EAC/LDPR,EALUC/A+B,AMX/PR,BMXC/#80,
;2473 ;  FPSYNC,NAD/0F9
;2474 ; 0F9: WAIT,NAD/0EF
;2475 ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
;2476 ; 0F7: BSC/NH,NAD/028
;2477 ; 028: NAD/028
;2478 ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,AMXC/LA,NAD/0F6
;2479 ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
;2480 ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/027
;2481 ; 027: BSC/R,SCR/CPU,EAC/LDAB,WAIT,NAD/02B
;2482 ;
;2483 ; LOGIC DESCRIPTION
;2484 ;
;2485 ; ALL OF THE  LOGIC IS ON THE FCT MODULE.
;2486 ;
;2487 ; ERROR DESCRIPTION
;2488 ; EXPECTED SIGN RESULT
;2489 ; RECEIVED SIGN RESULT
;2490 ;
;2491 ; SYNC POINT DESCRIPTION
;2492 ;
;2493 ;--
;2494 ;*****************************************************************
;2495 ;/////////////////////////////////////////////
;*
;2496 ;/////////////////////////////////////////////
;*
;2497 ; SUBTEST 1
;2498 ; [(EALU<3:0>=0H)+(EALU<7:4>=0H)+(-EALU8L)]*(-EALU9H)
;2499 ;-
;2500 =0
U 1094, 0018.0039.0980.09F8.0000.1181  ;2501 SGN3: NEWTST[.2]
U 1095, 0000.003C.0180.0800.0000.1096  ;2502  NOP
U 1096, 0000.003D.0180.0800.0000.1198  ;2503 =0 SUBTEST
U 1097, 0003.003C.0180.09F0.0000.1098  ;2504  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1098, 0000.003D.0180.0800.0000.11CF  ;2505 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 1099, 0818.0038.4580.0800.0000.1213  ;2506  D_K[.8000]
U 1213, 0001.003C.0180.0AA8.0000.109A  ;2507  R[5]_D  ; LA=0,SA=1
U 109A, 0000.003D.0180.0800.0000.11D7  ;2508 =0 CALL[LD.PR]  ; SET EALU <7:0>  EQUAL TO ZERO
U 109B, 0000.003C.0180.0800.0000.109C  ;2509  NOP
```

```
U 109C, 2018,0039,0180,0800,4200,11B7  ;2510 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 109D, 0000,003C,0180,0800,0000,109E  ;2511  NOP
U 109E, 0000,003D,0180,0800,0000,11B5  ;2512 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 109F, 0018,0038,C980,0AA8,0000,10A0  ;2513  R[5]_K[.3030]
U 10A0, 0000,003D,0180,0800,0000,11D7  ;2514 =0 CALL[LD.PR]  ; SET EALU<7:0> NOT EQUAL TO ZERO
U 10A1, 0000,003C,0180,0800,0000,10A2  ;2515  NOP
U 10A2, 0000,003D,0180,0800,0000,11B2  ;2516 =0 CALL[TRP.F7]  ; SA_SA
U 10A3, 0000,003C,01D8,0800,0000,1214  ;2517  Q_ACC
U 1214, 0019,2034,45C0,0800,0000,1215  ;2518  Q_Q.AND.K[.8000]
U 1215, 0001,203C,0180,09E8,0000,1216  ;2519  RC[0D]_Q
U 1216, 0011,2020,0180,0970,0010,1217  ;2520  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1217, 0000,013C,0180,0800,0000,10A4  ;2521  Z?
U 10A4, 0000,003D,0180,0800,0000,1194  ;2522 =0 ERROR1
U 10A5, 0000,003C,0180,0800,0000,10A6  ;2523  NOP
            ;2524 ;////////////////////////////////////////////
            ;2525 ;*
            ;2526 ; SUBTEST 2
            ;2527 ; [(EALU<3:0>=0H)*(EALU<7:4>=0H)*(EALU8L)]*(-EALU9H)
            ;2528 ;-
U 10A6, 0000,003D,0180,0800,0000,1198  ;2529 =0 SUBTEST
U 10A7, 0000,003C,0180,0800,0000,10A8  ;2530  NOP
U 10A8, 0000,003D,0180,0800,0000,1193  ;2531 =0 ERLOOP
U 10A9, 0000,003C,0180,0800,0000,1218  ;2532  NOP
U 1218, 0018,0038,4580,09F0,0000,10AA  ;2533  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10AA, 0000,003D,0180,0800,0000,11CF  ;2534 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 10AB, 0818,0038,4580,0800,0000,1219  ;2535  D_K[.8000]
U 1219, 0001,003C,0180,0AA8,0000,10AC  ;2536  R[5]_D  ; LA=0,SA=1
U 10AC, 0000,003D,0180,0800,0000,11D7  ;2537 =0 CALL[LD.PR]  ; SET EALU <7:0>  EQUAL TO ZERO
U 10AD, 0000,003C,0180,0800,0000,10AE  ;2538  NOP
U 10AE, 2018,0039,0180,0800,4200,11B7  ;2539 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 10AF, 0000,003C,0180,0800,0000,10B0  ;2540  NOP
U 10B0, 0000,003D,0180,0800,0000,11B5  ;2541 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10B1, 0000,003C,0180,0800,0000,10B2  ;2542  NOP  ; SA_BUSA(15),SB_BUSB(15),
            ;2543     ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 10B2, 0818,0039,CD80,0800,0000,11A1  ;2544 =0 D_K[.F0],CALL[TRAP1] ; EALU8H ASSERTED
U 10B3, 0000,007C,0180,0800,0000,121A  ;2545  CPSYNC
U 121A, 0000,003C,01D8,0800,0000,121B  ;2546  Q_ACC
U 121B, 0019,2034,45C0,0800,0000,121C  ;2547  Q_Q.AND.K[.8000]
U 121C, 0001,203C,0180,09E8,0000,121D  ;2548  RC[0D]_Q
U 121D, 0011,2020,0180,0970,0010,121E  ;2549  ALU_Q.XOR.RC[0E],CLK.UBCC
U 121E, 0000,013C,0180,0800,0000,10B4  ;2550  Z?
U 10B4, 0000,003D,0180,0800,0000,1194  ;2551 =0 ERROR1
U 10B5, 0000,003C,0180,0800,0000,10B6  ;2552  NOP
            ;2553 ;////////////////////////////////////////////
            ;2554 ;*
            ;2555 ; SUBTEST 3
            ;2556 ; [(-EALU<3:0>=0H)*(EALU<7:4>=0H)*(-EALU8L)]*(-EALU9H)
            ;2557 ;-
U 10B6, 0000,003D,0180,0800,0000,1198  ;2558 =0 SUBTEST
U 10B7, 0000,003C,0180,0800,0000,10B8  ;2559  NOP
U 10B8, 0000,003D,0180,0800,0000,1193  ;2560 =0 ERLOOP
U 10B9, 0000,003C,0180,0800,0000,121F  ;2561  NOP
U 121F, 0018,0038,4580,09F0,0000,10BA  ;2562  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10BA, 0000,003D,0180,0800,0000,11C3  ;2563 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 10BB, 0800,003C,0180,0A30,0000,1220  ;2564  D_R[6]
```

```
U 1220. 0019.0030.4580.0AB0.0000.10BC  ;2565  R[6]_D.OR.K[.8000]  ; SB=1
U 10BC. 0000.003D.0180.0800.0000.11D7  ;2566 =0 CALL[LD.PR]  ; SET EALU <3:0> NOT EQUAL TO ZERO
          ;2567     ; SET EALU <7:4> EQUAL ZERO
U 10BD. 0000.003C.0180.0800.0000.10BE  ;2568  NOP
U 10BE. 2018.0039.0180.0800.4200.11B7  ;2569 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 10BF. 0000.003C.0180.0800.0000.10C0  ;2570  NOP
U 10C0. 0000.003D.0180.0800.0000.11B5  ;2571 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10C1. 0000.007C.0180.0800.0000.1221  ;2572  CPSYNC
U 1221. 0000.003C.01D8.0800.0000.1222  ;2573  Q_ACC
U 1222. 0019.2034.45C0.0800.0000.1223  ;2574  Q_Q.AND.K[.8000]
U 1223. 0001.203C.0180.09E8.0000.1224  ;2575  RC[0D]_Q
U 1224. 0011.2020.0180.0970.0010.1225  ;2576  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1225. 0000.013C.0180.0800.0000.10C2  ;2577  Z?
U 10C2. 0000.003D.0180.0800.0000.1194  ;2578 =0 ERROR1
U 10C3. 0000.003C.0180.0800.0000.10C4  ;2579  NOP
          ;2580 ;////////////////////////////////////////////
          ;2581 ;*
          ;2582 ; SUBTEST 4
          ;2583 ; [(EALU<3:0>=0H)*(-EALU<7:4>=0H)*(-EALU8L)]*(-EALU9H)
          ;2584 ;-
U 10C4. 0000.003D.0180.0800.0000.1198  ;2585 =0 SUBTEST
U 10C5. 0000.003C.0180.0800.0000.10C6  ;2586  NOP
U 10C6. 0000.003D.0180.0800.0000.1193  ;2587 =0 ERLOOP
U 10C7. 0000.003C.0180.0800.0000.1226  ;2588  NOP
U 1226. 0018.0038.4580.09F0.0000.10C8  ;2589  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10C8. 0000.003D.0180.0800.0000.11CF  ;2590 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 10C9. 0818.0038.C980.0800.0000.1227  ;2591  D_K[.3030]
U 1227. 0019.0024.4980.0AA8.0000.10CA  ;2592  R[5]_ALU,ALU_D.ANDNOT.K[.FF] ; LA<7:4> NOT EQUAL ZERO
U 10CA. 0000.003D.0180.0800.0000.11D7  ;2593 =0 CALL[LD.PR]  ; SET EALU <7:4> NOT EQUAL TO ZERO
          ;2594     ; SET EALU <3:0> EQUAL TO ZERO
U 10CB. 0000.003C.0180.0800.0000.10CC  ;2595  NOP
U 10CC. 2018.0039.0180.0800.4200.11B7  ;2596 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 10CD. 0000.003C.0180.0800.0000.10CE  ;2597  NOP
U 10CE. 0000.003D.0180.0800.0000.11B5  ;2598 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10CF. 0000.007C.0180.0800.0000.1228  ;2599  CPSYNC
U 1228. 0000.003C.01D8.0800.0000.1229  ;2600  Q_ACC
U 1229. 0019.2034.45C0.0800.0000.122A  ;2601  Q_Q.AND.K[.8000]
U 122A. 0001.203C.0180.09E8.0000.122B  ;2602  RC[0D]_Q
U 122B. 0011.2020.0180.0970.0010.122C  ;2603  ALU_Q.XOR.RC[0E],CLK.UBCC
U 122C. 0000.013C.0180.0800.0000.10D0  ;2604  Z?
U 10D0. 0000.003D.0180.0800.0000.1194  ;2605 =0 ERROR1
U 10D1. 0000.003C.0180.0800.0000.10D2  ;2606  NOP
          ;2607 ;////////////////////////////////////////////
;*
          ;2608 ;////////////////////////////////////////////
;*
          ;2609 ; SUBTEST 5
          ;2610 ; [(EALU<3:0>=0H)*(EALU<7:4>=0H)*(EALU8L)]*(EALU9H)
          ;2611 ;-
U 10D2. 0000.003D.0180.0800.0000.1198  ;2612 =0 SUBTEST
U 10D3. 0000.003C.0180.0800.0000.10D4  ;2613  NOP
U 10D4. 0000.003D.0180.0800.0000.1193  ;2614 =0 ERLOOP
U 10D5. 0000.003C.0180.0800.0000.122D  ;2615  NOP
U 122D. 0003.003C.0180.09F0.0000.10D6  ;2616  RC[0E]_0  ; EXPECTED SIGN RESULT
U 10D6. 0000.003D.0180.0800.0000.11CF  ;2617 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 10D7. 0818.0038.4580.0800.0000.122E  ;2618  D_K[.8000]
U 122E. 0001.003C.0180.0AA8.0000.10D8  ;2619  R[5]_D  ; LA=0,SA=1
```

```
U 10D8, 0000,003D,0180,0800,0000,11D7  ;2620 =0 CALL[LD.PR]  ; SET EALU <7:0>  EQUAL TO ZERO
U 10D9, 0000,003C,0180,0800,0000,10DA  ;2621  NOP
U 10DA, 2018,0039,0180,0800,4200,11B7  ;2622 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 10DB, 0000,003C,0180,0800,0000,10DC  ;2623  NOP
U 10DC, 0000,003D,0180,0800,0000,11B5  ;2624 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 10DD, 0000,003C,0180,0800,0000,10DE  ;2625  NOP  ; SA_BUSA(15),SB_BUSB(15),
          ;2626    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 10DE, 0818,0039,CD80,0800,0000,11A1  ;2627 =0 D_K[.F0],CALL[TRAP1]  ; PR_LA
U 10DF, 0000,007C,0180,0800,0000,10E0  ;2628  CPSYNC
U 10E0, 0000,003D,1180,0800,1404,71A4  ;2629 =0 STATE_K[.4],CALL[STEP] ; EALU8H AND EALU9H ASSERTED
U 10E1, 0000,007C,0180,0800,0000,122F  ;2630  CPSYNC
U 122F, 0000,003C,01D8,0800,0000,1230  ;2631  Q_ACC
U 1230, 0019,2034,45C0,0800,0000,1231  ;2632  Q_Q.AND.K[.8000]
U 1231, 0001,203C,0180,09E8,0000,1232  ;2633  RC[0D]_Q
U 1232, 0011,2020,0180,0970,0010,1233  ;2634  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1233, 0000,013C,0180,0800,0000,10E2  ;2635  Z?
U 10E2, 0000,003D,0180,0800,0000,1194  ;2636 =0 ERROR1
U 10E3, 0000,003C,0180,0800,0000,10E4  ;2637  NOP
```

```
                ;2638  .PAGE "TEST 27F SIGN LOGIC FOR ADD/SUB WITH LA>LB"
                ;2639 ;*************************************************************************
                ;2640 ;**
                ;2641 ; TEST SIGN LOGIC FOR ADD/SUB WITH LA>LB
                ;2642 ;
                ;2643 ; TEST DESCRIPTION
                ;2644 ;
                ;2645 ; THE SIGN DECISION LOGIC FOR ADD AND SUBTRACT, WITH
                ;2646 ; LA>LB, IS CHECKED IN THIS TEST. THE RESULTING SIGN
                ;2647 ; OUTPUT IS CHECKED WITH ONLY THE REQUIRED INPUT AND
                ;2648 ; MUX SELECT TO THE SIGN OUT PUT MUX ASSERTED. THEN
                ;2649 ; ALL POSSIBLE COMBINATIONS OF SIGN INPUT AND MUX
                ;2650 ; SELECT ARE ASSERTED, AND THE SIGN OUT PUT IS VERIFIED.
                ;2651 ;
                ;2652 ;
                ;2653 ; FPA UCODE USED
                ;2654 ; 0F1: BSC/NH,SGNC/A.SNA,NAD/PSTALL
                ;2655 ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
                ;2656 ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,AMXC/LA,NAD/0F6
                ;2657 ; 088: BSC/R,SCR/CPU,EA/LDB,SGNC/LDSB,FADC/BR1,NAD/PSTALL
                ;2658 ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
                ;2659 ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/027
                ;2660 ; 027: BSC/R,SCR/CPU,EAC/LDAB,WAIT,NAD/02B
                ;2661 ;
                ;2662 ; LOGIC DESCRIPTION
                ;2663 ; ALL OF THE LOGIC FOR THIS FUNCTION IS ON THE FCT MODULE.
                ;2664 ;
                ;2665 ; ERROR DESCRIPTION
                ;2666 ; EXPECTED SIGN RESULT
                ;2667 ; RECEIVED SIGN RESULT
                ;2668 ;
                ;2669 ; SYNC POINT DESCRIPTION
                ;2670 ;
                ;2671 ;--
                ;2672 ;*************************************************************************
                ;2673 ;/////////////////////////////////////////////
                ;2674 ;*
                ;2675 ; SUBTEST 1
                ;2676 ; (SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
                ;2677 ;-
                ;2678 =0
U 10E4, 0018.0039.0980.09F8.0000.1181  ;2679 SGN4: NEWTST[.2]
U 10E5, 0000.003C.0180.0800.0000.10E6  ;2680  NOP
U 10E6, 0000.003D.0180.0800.0000.1198  ;2681 =0 SUBTEST
U 10E7, 0000.003C.0180.0800.0000.1234  ;2682  NOP
U 1234, 0018.0038.4580.09F0.0000.10E8  ;2683  RC[0E] K[.8000]  ; EXPECTED SIGN RESULT
U 10E8, 0000.003D.0180.0800.0000.11C0  ;2684 =0 CALL[LDDAT1]  ; LA>LB,SA=0,SB=1
U 10E9, 0000.003C.0180.0800.0000.10EA  ;2685  NOP
U 10EA, 0000.003D.0180.0800.0000.11D7  ;2686 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10EB, 0000.003C.0180.0800.0000.10EC  ;2687  NOP
U 10EC, 2018.0039.8580.0800.4200.11B7  ;2688 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS SUBF R,R
U 10ED, 0000.003C.0180.0800.0000.10EE  ;2689  NOP
U 10EE, 0000.003D.0180.0800.0000.11B5  ;2690 =0 CALL[TRP.F8]
U 10EF, 0000.007C.0180.0800.0000.1235  ;2691  CPSYNC
U 1235, 0000.003C.01D8.0800.0000.1236  ;2692  Q_ACC
```

```
U 1236. 0019.2034.45C0.0800.0000.1237  ;2693  Q_Q.AND.K[.8000]
U 1237. 0001.203C.0180.09E8.0000.1238  ;2694  RC[0D]_Q
U 1238. 0011.2020.0180.0970.0010.1239  ;2695  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1239. 0000.013C.0180.0800.0000.10F0  ;2696  Z?
U 10F0. 0000.003D.0180.0800.0000.1194  ;2697 =0 ERROR1
U 10F1. 0000.003C.0180.0800.0000.10F2  ;2698  NOP
          ;2699 ;/////////////////////////////////////////////
          ;2700 ;*
          ;2701 ; SUBTEST 2
          ;2702 ; (-SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
          ;2703 ;-
U 10F2. 0000.003D.0180.0800.0000.1198  ;2704 =0 SUBTEST
U 10F3. 0000.003C.0180.0800.0000.10F4  ;2705  NOP
U 10F4. 0000.003D.0180.0800.0000.1193  ;2706 =0 ERLOOP
U 10F5. 0000.003C.0180.0800.0000.123A  ;2707  NOP
U 123A. 0003.003C.0180.09F0.0000.10F6  ;2708  RC[0E]_0   ; EXPECTED SIGN RESULT
U 10F6. 0000.003D.0180.0800.0000.11C8  ;2709 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 10F7. 0000.003C.0180.0800.0000.10F8  ;2710  NOP
U 10F8. 0000.003D.0180.0800.0000.11D7  ;2711 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 10F9. 0000.003C.0180.0800.0000.10FA  ;2712  NOP
U 10FA. 2018.0039.8580.0800.4200.11B7  ;2713 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS SUBF R,R
U 10FB. 0000.003C.0180.0800.0000.10FC  ;2714  NOP
U 10FC. 0000.003D.0180.0800.0000.11B5  ;2715 =0 CALL[TRP.F8]
U 10FD. 0000.007C.0180.0800.0000.123B  ;2716  CPSYNC
U 123B. 0000.003C.01D8.0800.0000.123C  ;2717  Q_ACC
U 123C. 0019.2034.45C0.0800.0000.123D  ;2718  Q_Q.AND.K[.8000]
U 123D. 0001.203C.0180.09E8.0000.123E  ;2719  RC[0D]_Q
U 123E. 0011.2020.0180.0970.0010.123F  ;2720  ALU_Q.XOR.RC[0E],CLK.UBCC
U 123F. 0000.013C.0180.0800.0000.10FE  ;2721  Z?
U 10FE. 0000.003D.0180.0800.0000.1194  ;2722 =0 ERROR1
U 10FF. 0000.003C.0180.0800.0000.110A  ;2723  NOP
          ;2724 ;/////////////////////////////////////////////
          ;2725 ;*
          ;2726 ; SUBTEST 3
          ;2727 ; (SA(0)H.XOR.-IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
          ;2728 ;-
U 110A. 0000.003D.0180.0800.0000.1198  ;2729 =0 SUBTEST
U 110B. 0000.003C.0180.0800.0000.1110  ;2730  NOP
U 1110. 0000.003D.0180.0800.0000.1193  ;2731 =0 ERLOOP
U 1111. 0000.003C.0180.0800.0000.1240  ;2732  NOP
U 1240. 0003.003C.0180.09F0.0000.1112  ;2733  RC[0E]_0   ; EXPECTED SIGN RESULT
U 1112. 0000.003D.0180.0800.0000.11C0  ;2734 =0 CALL[LDDAT1]  ; LA>LB,SA=0,SB=1
U 1113. 0000.003C.0180.0800.0000.1114  ;2735  NOP
U 1114. 0000.003D.0180.0800.0000.11D7  ;2736 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1115. 0000.003C.0180.0800.0000.1116  ;2737  NOP
U 1116. 2018.0039.0180.0800.4200.11B7  ;2738 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1117. 0000.003C.0180.0800.0000.1118  ;2739  NOP
U 1118. 0000.003D.0180.0800.0000.11B5  ;2740 =0 CALL[TRP.F8]
U 1119. 0000.007C.0180.0800.0000.1241  ;2741  CPSYNC
U 1241. 0000.003C.01D8.0800.0000.1242  ;2742  Q_ACC
U 1242. 0019.2034.45C0.0800.0000.1243  ;2743  Q_Q.AND.K[.8000]
U 1243. 0001.203C.0180.09E8.0000.1244  ;2744  RC[0D]_Q
U 1244. 0011.2020.0180.0970.0010.1245  ;2745  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1245. 0000.013C.0180.0800.0000.111A  ;2746  Z?
U 111A. 0000.003D.0180.0800.0000.1194  ;2747 =0 ERROR1
```

```
U 111B. 0000.003C.0180.0800.0000.111C  ;2748  NOP
        ;2749 ;////////////////////////////////////////
        ;2750 ;*
        ;2751 ; SUBTEST 4
        ;2752 ; (SA(0)H.XOR.IR1L)*(A<B)*[(-POLYADDL)+(-IR2H)]*(-A OU OF NORM L)
        ;2753 ;-
U 111C. 0000.003D.0180.0800.0000.1198  ;2754 =0 SUBTEST
U 111D. 0000.003C.0180.0800.0000.111E  ;2755  NOP
U 111E. 0000.003D.0180.0800.0000.1193  ;2756 =0 ERLOOP
U 111F. 0000.003C.0180.0800.0000.1246  ;2757  NOP
U 1246. 0003.003C.0180.09F0.0000.1130  ;2758  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1130. 0000.003D.0180.0800.0000.11C3  ;2759 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 1131. 0000.003C.0180.0800.0000.1132  ;2760  NOP
U 1132. 0000.003D.0180.0800.0000.11D7  ;2761 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1133. 0000.003C.0180.0800.0000.1134  ;2762  NOP
U 1134. 2018.0039.8580.0800.4200.11B7  ;2763 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS SUBF R,R
U 1135. 0000.003C.0180.0800.0000.1136  ;2764  NOP
U 1136. 0000.003D.0180.0800.0000.11B5  ;2765 =0 CALL[TRP.F8]
U 1137. 0000.007C.0180.0800.0000.1247  ;2766  CPSYNC
U 1247. 0000.003C.01D8.0800.0000.1248  ;2767  Q_ACC
U 1248. 0019.2034.45C0.0800.0000.1249  ;2768  Q_Q.AND.K[.8000]
U 1249. 0001.203C.0180.09E8.0000.124A  ;2769  RC[0D]_Q
U 124A. 0011.2020.0180.0970.0010.124B  ;2770  ALU_Q.XOR.RC[0E],CLK.UBCC
U 124B. 0000.013C.0180.0800.0000.1138  ;2771  Z?
U 1138. 0000.003D.0180.0800.0000.1194  ;2772 =0 ERROR1
U 1139. 0000.003C.0180.0800.0000.113A  ;2773  NOP
        ;2774 ;////////////////////////////////////////
        ;2775 ;*
        ;2776 ; SUBTEST 5
        ;2777 ; (-SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
        ;2778 ;-
U 113A. 0000.003D.0180.0800.0000.1198  ;2779 =0 SUBTEST
U 113B. 0000.003C.0180.0800.0000.113C  ;2780  NOP
U 113C. 0000.003D.0180.0800.0000.1193  ;2781 =0 ERLOOP
U 113D. 0000.003C.0180.0800.0000.124C  ;2782  NOP
U 124C. 0018.0038.4580.09F0.0000.113E  ;2783  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 113E. 0000.003D.0180.0800.0000.11C8  ;2784 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 113F. 0000.003C.0180.0800.0000.1140  ;2785  NOP
U 1140. 0000.003D.0180.0800.0000.11D7  ;2786 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1141. 0000.003C.0180.0800.0000.1142  ;2787  NOP
U 1142. 2018.0039.0180.0800.4200.11B7  ;2788 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1143. 0000.003C.0180.0800.0000.1144  ;2789  NOP
U 1144. 0000.003D.0180.0800.0000.11B5  ;2790 =0 CALL[TRP.F8]
U 1145. 0000.007C.0180.0800.0000.124D  ;2791  CPSYNC
U 124D. 0000.003C.01D8.0800.0000.124E  ;2792  Q_ACC
U 124E. 0019.2034.45C0.0800.0000.124F  ;2793  Q_Q.AND.K[.8000]
U 124F. 0001.203C.0180.09E8.0000.1250  ;2794  RC[0D]_Q
U 1250. 0011.2020.0180.0970.0010.1251  ;2795  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1251. 0000.013C.0180.0800.0000.1146  ;2796  Z?
U 1146. 0000.003D.0180.0800.0000.1194  ;2797 =0 ERROR1
U 1147. 0000.003C.0180.0800.0000.1148  ;2798  NOP
        ;2799 ;////////////////////////////////////////
        ;2800 ;*
        ;2801 ; SUBTEST 6
        ;2802 ; (SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
```

```
            ;2803 ;-
U 1148. 0000.003D.0180.0800.0000.1198  ;2804 =0 SUBTEST
U 1149. 0000.003C.0180.0800.0000.114A  ;2805  NOP
U 114A. 0000.003D.0180.0800.0000.1193  ;2806 =0 ERLOOP
U 114B. 0000.003C.0180.0800.0000.1252  ;2807  NOP
U 1252. 0003.003C.0180.09F0.0000.114C  ;2808  RC[0E]_0  ; EXPECTED SIGN RESULT
U 114C. 0000.003D.0180.0800.0000.11C0  ;2809 =0 CALL[LDDAT1]  ; LA>LB,SA=0,SB=1
U 114D. 0000.003C.0180.0800.0000.114E  ;2810  NOP
U 114E. 0000.003D.0180.0800.0000.11D7  ;2811 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 114F. 0000.003C.0180.0800.0000.1150  ;2812  NOP
U 1150. 2018.0039.0180.0800.4200.11B7  ;2813 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1151. 0000.003C.0180.0800.0000.1152  ;2814  NOP
U 1152. 0000.003D.0180.0800.0000.11B5  ;2815 =0 CALL[TRP.F8]
U 1153. 0000.003C.0180.0800.0000.1156  ;2816  NOP  ; SA_BUSA(15),SB_BUSB(15),
            ;2817    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 1156. 0000.003D.0180.0800.0000.11AC  ;2818 =0 CALL[TRP.F1]  ; SA_COM
U 1157. 0000.003C.01D8.0800.0000.1253  ;2819  Q_ACC
U 1253. 0019.2034.45C0.0800.0000.1254  ;2820  Q_Q.AND.K[.8000]
U 1254. 0001.203C.0180.09E8.0000.1255  ;2821  RC[0D]_Q
U 1255. 0011.2020.0180.0970.0010.1256  ;2822  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1256. 0000.013C.0180.0800.0000.1158  ;2823  Z?
U 1158. 0000.003D.0180.0800.0000.1194  ;2824 =0 ERROR1
U 1159. 0000.003C.0180.0800.0000.115A  ;2825  NOP
            ;2826 ;////////////////////////////////////////////
            ;2827 ;*
            ;2828 ; SUBTEST 7
            ;2829 ; (-SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
            ;2830 ; TEST FOR SA_COM
            ;2831 ;-
U 115A. 0000.003D.0180.0800.0000.1198  ;2832 =0 SUBTEST
U 115B. 0000.003C.0180.0800.0000.115C  ;2833  NOP
U 115C. 0000.003D.0180.0800.0000.1193  ;2834 =0 ERLOOP
U 115D. 0000.003C.0180.0800.0000.1257  ;2835  NOP
U 1257. 0018.0038.4580.09F0.0000.115E  ;2836  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 115E. 0000.003D.0180.0800.0000.11C8  ;2837 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 115F. 0000.003C.0180.0800.0000.1160  ;2838  NOP
U 1160. 0000.003D.0180.0800.0000.11D7  ;2839 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1161. 0000.003C.0180.0800.0000.1162  ;2840  NOP
U 1162. 2018.0039.0180.0800.4200.11B7  ;2841 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1163. 0000.003C.0180.0800.0000.1164  ;2842  NOP
U 1164. 0000.003D.0180.0800.0000.11B5  ;2843 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1165. 0000.003C.0180.0800.0000.1166  ;2844  NOP  ; SA_BUSA(15),SB_BUSB(15),
            ;2845    ; LA_BUSA<14:07>,LB_BUSB<14:07>
U 1166. 0000.003D.0180.0800.0000.11AC  ;2846 =0 CALL[TRP.F1]  ; SA_COM
U 1167. 0000.003C.01D8.0800.0000.1258  ;2847  Q_ACC
U 1258. 0019.2034.45C0.0800.0000.1259  ;2848  Q_Q.AND.K[.8000]
U 1259. 0001.203C.0180.09E8.0000.125A  ;2849  RC[0D]_Q
U 125A. 0011.2020.0180.0970.0010.125B  ;2850  ALU_Q.XOR.RC[0E],CLK.UBCC
U 125B. 0000.013C.0180.0800.0000.1168  ;2851  Z?
U 1168. 0000.003D.0180.0800.0000.1194  ;2852 =0 ERROR1
U 1169. 0000.003C.0180.0800.0000.116A  ;2853  NOP
            ;2854 ;////////////////////////////////////////////
            ;2855 ;*
            ;2856 ; SUBTEST 8
            ;2857 ; (-SA(0)H.XOR.IR1L)*(A>B)*[(-POLYADDL)+(IR2H)]*(-A OUT OF NORM L)
```

```
            ;2858 ; TEST FOR SA_COM
            ;2859 ;-
U 116A. 0000.003D.0180.0800.0000.1198  ;2860 =0 SUBTEST
U 116B. 0000.003C.0180.0800.0000.116C  ;2861  NOP
U 116C. 0000.003D.0180.0800.0000.1193  ;2862 =0 ERLOOP
U 116D. 0000.003C.0180.0800.0000.125C  ;2863  NOP
U 125C. 0003.003C.0180.09F0.0000.116E  ;2864  RC[0E]_0  ; EXPECTED SIGN RESULT
U 116E. 0000.003D.0180.0800.0000.11C8  ;2865 =0 CALL[LDDAT4]  ; LA>LB,SA=1,SB=1
U 116F. 0000.003C.0180.0800.0000.1170  ;2866  NOP
U 1170. 0000.003D.0186.0800.0000.11D7  ;2867 =0 CALL[LD.PR]  ; SET EALU <7:0> NOT EQUAL TO ZERO
U 1171. 0000.003C.0180.0800.0000.1172  ;2868  NOP
U 1172. 2018.0039.2180.0800.4200.11B7  ;2869 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS MULF R,R
U 1173. 0000.003C.0180.0800.0000.1174  ;2870  NOP
U 1174. 0000.003D.0180.0800.0000.11B5  ;2871 =0 CALL[TRP.F8]
U 1175. 0000.007C.0180.0800.0000.125D  ;2872  CPSYNC
U 125D. 0000.003C.01D8.0800.0000.125E  ;2873  Q_ACC
U 125E. 0019.2034.45C0.0800.0000.125F  ;2874  Q_Q.AND.K[.8000]
U 125F. 0001.203C.0180.09E8.0000.1260  ;2875  RC[0D]_Q
U 1260. 0011.2020.0180.0970.0010.1261  ;2876  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1261. 0000.013C.0180.0800.0000.1176  ;2877  Z?
U 1176. 0000.003D.0180.0800.0000.1194  ;2878 =0 ERROR1
U 1177. 0000.003C.0180.0800.0000.1262  ;2879  NOP
U 1262. 0011.2020.0180.0970.0010.1263  ;2880  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1263. 0000.013C.0180.0800.0000.1178  ;2881  Z?
U 1178. 0000.003D.0180.0800.0000.1194  ;2882 =0 ERROR1
U 1179. 0000.003C.0180.0800.0000.117A  ;2883  NOP
            ;2884
```

```
ZZ-ESKAR-V22  TEST 280 RESERVED                                                    SEQ 2651
ESKAR61.MCR                        MICRO2  1P(00)     26-JUL-85  17:06:17         Page   7

            ;2885  .PAGE "TEST 280 RESERVED"
            ;2886 =0
U 117A, 0000,003D,0180,0800,0000,1181  ;2887 TST12K: NEWTST
U 117B, 0000,003C,0180,0800,0000,117C  ;2888  NOP
            ;2889
```

```
            ;2890  .PAGE "DUMMY NEWTST"
            ;2891 =0
U 117C. 0000,003D,0180,0800,0000,1181  ;2892 END12: NEWTST
            ;2893
            ;2894
            ;2895 ; THE FOLLOWING OPCODE DATA IS REQUIRED FOR THIS TEST.
            ;2896 ;* 8
            ;2897 ;$ 5540. 0056
            ;2898 ;$ 5542. 0056
            ;2899 ;$ 5546. 0056
            ;2900 ;$ 5544. 0056
            ;2901 ;$ 5A40. 005B
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 1918= 1919=
U 1008  2015= 2016= 1875 2000= 2094= 2095= 1876 2001=
U 1010  2097= 2098= 2103= 2104= 2106= 2107= 2113= 2114=
U 1018  2116= 2117= 1877 2028= 2123= 2124= 1878 2029=
U 1020  2126= 2127= 2134= 2135= 2138= 2139= 2145= 2146=
U 1028  2154= 2155= 2158= 2159= 2183= 2184= 2204= 2205=
U 1030  2207= 2208= 2209= 2210= 2212= 2213= 2226= 2227=
U 1038  2307= 2308= 2309= 2310= 2311= 2312= 2313= 2314=
U 1040  2315= 2316= 2317= 2318= 2324= 2325= 2331= 2332=
U 1048  2333= 2334= 2336= 2337= 2339= 2340= 1939: 1879
U 1050  2341= 2342= 2343= 2344= 2350= 2351= 2357= 2358=
U 1058  2359= 2360= 1942: 1880 2362= 2363= 1945: 1900
U 1060  2364= 2365= 2366= 2367= 2368= 2369= 2375= 2376=
U 1068  2382= 2383= 2384= 2385= 2387= 2388= 2389= 2390=
U 1070  2391= 2392= 2393= 2394= 2400= 2401= 2407= 2408=
U 1078  2409= 2410= 2412= 2413= 2414= 2415= 2416= 2417=
U 1080  2418= 2419= 2425= 2426= 2432= 2433= 2434= 2435=
U 1088  2437= 2438= 2439= 2440= 2441= 2442= 2443= 2444=
U 1090  2446= 2447= 2453= 2454= 2501= 2502= 2503= 2504=
U 1098  2505= 2506= 2508= 2509= 2510= 2511= 2512= 2513=
U 10A0  2514= 2515= 2516= 2517= 2522= 2523= 2529= 2530=
U 10A8  2531= 2532= 2534= 2535= 2537= 2538= 2539= 2540=
U 10B0  2541= 2542= 2544= 2545= 2551= 2552= 2558= 2559=
U 10B8  2560= 2561= 2563= 2564= 2566= 2568= 2569= 2570=
U 10C0  2571= 2572= 2578= 2579= 2585= 2586= 2587= 2588=
U 10C8  2590= 2591= 2593= 2595= 2596= 2597= 2598= 2599=
U 10D0  2605= 2606= 2612= 2613= 2614= 2615= 2617= 2618=
U 10D8  2620= 2621= 2622= 2623= 2624= 2625= 2627= 2628=
U 10E0  2629= 2630= 2636= 2637= 2679= 2680= 2681= 2682=
U 10E8  2684= 2685= 2686= 2687= 2688= 2689= 2690= 2691=
U 10F0  2697= 2698= 2704= 2705= 2706= 2707= 2709= 2710=
U 10F8  2711= 2712= 2713= 2714= 2715= 2716= 2722= 2723=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1901 2729= 2730= 1870: 1871: 1872: 1873:
U 1110  2731= 2732= 2734= 2735= 2736= 2737= 2738= 2739=
U 1118  2740= 2741= 2747= 2748= 2754= 2755= 2756= 2757=
U 1120  2251: 2252: 2253: 2254: 2255: 2256: 2257: 2258:
U 1128  2259: 2260: 2261: 2262: 2263: 2264: 2265: 2266:
U 1130  2759= 2760= 2761= 2762= 2763= 2764= 2765= 2766=
U 1138  2772= 2773= 2779= 2780= 2781= 2782= 2784= 2785=
U 1140  2786= 2787= 2788= 2789= 2790= 2791= 2797= 2798=
U 1148  2804= 2805= 2806= 2807= 2809= 2810= 2811= 2812=
U 1150  2813= 2814= 2815= 2816= 1902 1965: 2818= 2819=
U 1158  2824= 2825= 2832= 2833= 2834= 2835= 2837= 2838=
U 1160  2839= 2840= 2841= 2842= 2843= 2844= 2846= 2847=
U 1168  2852= 2853= 2860= 2861= 2862= 2863= 2865= 2866=
U 1170  2867= 2868= 2869= 2870= 2871= 2872= 2878= 2879=
U 1178  2882= 2883= 2887= 2888= 2892= 1903 1904 1905
U 1180  1906 1916 1917 1920 1921 1922 1923 1924
U 1188  1925 1926 1927 1928 1929 1930 1931 1932
U 1190  1933 1934 1935 1936 1937 1938 1940 1941
U 1198  1973 1974 1980 1981 1982 1996 1997 1998
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  1999 2013 2014 2017 2025 2026 2027 2040
U 11A8  2041 2044 2045 2046 2052 2053 2056 2057
U 11B0  2060 2061 2065 2066 2067 2070 2071 2080
U 11B8  2081 2082 2083 2084 2085 2086 2093 2096
U 11C0  2102 2105 2108 2112 2115 2122 2125 2128
U 11C8  2133 2136 2137 2140 2144 2147 2148 2152
U 11D0  2153 2156 2157 2160 2170 2171 2172 2181
U 11D8  2182 2185 2186 2187 2188 2202 2203 2206
U 11E0  2211 2214 2228 2229 2230 2236 2237 2238
U 11E8  2239 2240 2241 2242 2243 2244 2245 2319
U 11F0  2320 2321 2322 2323 2335 2338 2345 2346
U 11F8  2347 2348 2349 2361 2370 2371 2372 2373
U 1200  2374 2386 2395 2396 2397 2398 2399 2411
U 1208  2420 2421 2422 2423 2424 2436 2448 2449
U 1210  2450 2451 2452 2507 2518 2519 2520 2521
U 1218  2533 2536 2546 2547 2548 2549 2550 2562
U 1220  2565 2573 2574 2575 2576 2577 2589 2592
U 1228  2600 2601 2602 2603 2604 2616 2619 2631
U 1230  2632 2633 2634 2635 2683 2692 2693 2694
U 1238  2695 2696 2708 2717 2718 2719 2720 2721
U 1240  2733 2742 2743 2744 2745 2746 2758 2767
U 1248  2768 2769 2770 2771 2783 2792 2793 2794
U 1250  2795 2796 2808 2820 2821 2822 2823 2836
U 1258  2843 2849 2850 2851 2864 2873 2874 2875
U 1260  2876 2877 2880 2881
U 1268 - 12A7 Unused
U 12A8    1966:
U 12B0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR61	629
Used	629
Remaining	

Total microwords used in memory U: 629
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2656
ESKAR61.MCR                    MICRO2  1P(00)    26-JUL-85  17:06:17           Page   8


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR61.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
   ESKAR62.MCR   MICRO2 1P(00)   26-JUL-85 17 SEQ 2657

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2478 TEST 281 SIGN LOGIC FOR ADD/SUB WITH LA=LB
; 2779 TEST 282 SIGN LOGIC FOR ADD/SUB WITH LA<LB
; 3024 TEST 283 SIGN LOGIC FOR SA.XOR.SX
; 3165 TEST 284 SIGN LOGIC FOR POLYADDL AND ADD/SUB
; 3275 TEST 285 RESERVED
; 3280 TEST 286 RESERVED
; 3285 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR62
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100. 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003  ;1B6B 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D. 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.100A  ;1874 TRPH0: Q_ID[USTACK]
U 100A. 0C00.003C.01E0.0800.0000.100E  ;1875  Q_D,D_Q
U 100E. 0000.003C.8180.3C00.0000.104F  ;1876  ID[USTACK]_D
U 104F. 0C00.003C.01E0.0800.0000.105B  ;1877  Q_D,D_Q
U 105B. 0000.003C.01C0.0A78.0000.105F  ;1878  Q_R[0F]
U 105F. 0001.2024.0180.0800.0010.1109  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 1109. 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
                ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1004   ;1897 TRPH1: RC[0D]_SC
U 1004. 0000.003D.D980.0800.0084.7211   ;1898 =0 SETRCF[.9]
U 1005. 0000.003C.49F0.2C00.0000.1154   ;1899  Q_ID[TBER0]
U 1154. 0001.203C.4DF0.2DB0.0000.11F2   ;1900  RC[6]_Q,Q_ID[TBER1]
U 11F2. 0001.203C.65F0.2DB8.0000.11F3   ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 11F3. 0001.203C.6DF0.2DD8.0000.11F4   ;1902  RC[0B]_Q,Q_ID[FAULT]
U 11F4. 0001.203C.75F0.2DE0.0000.11F5   ;1903  RC[0C]_Q,Q_ID[MAINT]
U 11F5. 0001.203C.79F0.2DC8.0000.11F6   ;1904  RC[9]_Q,Q_ID[PARITY]
U 11F6. 0001.203C.69F0.2DC0.0000.11F7   ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 11F7. 0001.203C.81F0.2DD0.0000.1000   ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.120B   ;1907 1000: RC[0E]_Q,ERROR1
                ;1908 ;*
                ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1911 ;   NEWTST
                ;1912 ;   ERLOOP
                ;1913 ;   ERROR1
                ;1914 ;   ERROR2
                ;1915 ;
U 11F8. 0000.003C.8580.0800.0084.71F9   ;1916 SCOPE: SC_K[.C]
U 11F9. 0803.001C.0180.0800.0000.1006   ;1917  ALU_NOT.MASK,D_ALU
U 1006. 0000.003D.7580.3C00.0000.1262   ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1007. 0000.003C.65F8.0800.0084.71FA   ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 11FA. 0818.0038.8D80.0800.0000.11FB   ;1920  D_K[.1F]    ; ...
U 11FB. 0D00.003C.0180.0800.0000.11FC   ;1921  D_DAL.SC
U 11FC. 0000.003C.3D80.3C00.0000.11FD   ;1922  ID[PSL]_D    ; WRITE THE PSL
U 11FD. 0818.0038.0580.0800.0000.11FE   ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 11FE. 0D00.003C.0180.0800.0000.11FF   ;1924  D_DAL.SC
U 11FF. 0F00.003C.3180.3C00.0000.1200   ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 1200. 0000.003C.0180.0800.0000.1201   ;1926  NOP
U 1201. 0000.003C.3980.3C00.0000.1202   ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 1202. 0000.003C.2980.3C00.0000.1203   ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 1203. 0000.003C.4980.3C00.0000.1204   ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 1204. 0818.0038.C180.0800.0000.1205   ;1930  D_K[.FFFF]
U 1205. 0000.003C.6580.3C00.0000.1206   ;1931  ID[SBI.ERR]_D
U 1206. 0F00.003C.6D80.3C00.0000.1207   ;1932  ID[FAULT]_D,D_0
U 1207. 0000.003C.6D80.3C00.0000.1208   ;1933  ID[FAULT]_D
U 1208. 0000.003C.6580.3C00.0000.1209   ;1934  ID[SBI.ERR]_D
U 1209. 0003.003C.0180.0AF8.0000.105E   ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 120A. 0818.0038.0980.0800.0000.105E   ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 120B. 0810.0038.0180.0978.0000.120C   ;1937 ERROR1: D_RC[0F]
U 120C. 0819.0034.4D80.0800.0000.104E   ;1938  D_D.AND.K[.FF0D]
U 104E. 0819.0030.1180.0800.0000.105E   ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 120D. 0810.0038.0180.0978.0000.120E   ;1940 ERROR2: D_RC[0F]
U 120E. 0819.0034.4D80.0800.0000.105A   ;1941  D_D.AND.K[.FF0D]
U 105A. 0819.0030.D580.0800.0000.105E   ;1942 105A: D_D.OR.K[.6],J/CALLCON
                ;1943 105E:
                ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E   ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
                ;1946 ;*
                ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
                ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1   ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2   ;1950 13F1: NOP
```

```
U 13F2, 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3, 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4, 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5, 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6, 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7, 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8, 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9, 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA, 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB, 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC, 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD, 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE, 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF, 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155, 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
          ;1967 ;*
          ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST  NUMBER
          ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
          ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
          ;1971 ; TYPING IT.
          ;1972 ;-
U 120F, 0810.0038.4D80.0978.0000.1210  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 1210, 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D*K[.FF00],WORD,RETURN1
          ;1975 ;*
          ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
          ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
          ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
          ;1979 ;-
U 1211, 0010.0038.01C0.0978.0000.1212  ;1980 SETRCF: Q_RC[0F]
U 1212, 0019.2034.4DC0.0800.0000.1213  ;1981  Q_Q.AND.K[.FF00]
U 1213, 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
          ;1983
          ;1984
          ;1985
          ;1986
          ;1987
          ;1988
          ;1989 ;*****************************************************
          ;1990 ;
          ;1991 ; MAIN MACHINE/FPA SUBROUTINES
          ;1992 ;
          ;1993 ;*****************************************************
          ;1994
          ;1995 ; SHIFT D REGISTTER CONTENTS
          ;1996
          ;1997
          ;1998 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
          ;1999
          ;2000
U 1214, 0000.003C.01F8.0800.0000.1215  ;2001 SHIFTD: Q_0
U 1215, 0500.003C.0180.0800.0000.1216  ;2002  D_D.LEFT  ;
U 1216, 0000.003C.0580.0800.1414.B217  ;2003  EALU_STATE-K[.1],CLK.UBCC ;
U 1217, 0000.123C.0180.0800.0000.100B  ;2004  EALU.Z?
U 100B, 0000.003C.0180.0800.0000.1214  ;2005 =1011 J/SHIFTD  ;
```

```
U 100F, 0000,003E,0180,0800,0000,0001  ;2006  RETURN1
            ;2007
            ;2008 ;*
            ;2009 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
            ;2010 ; OR'S IN BIT 15,SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP
            ;2011 ; TO THAT ADDRESS.
            ;2012 ;-
            ;2013
U 1218, 0819,0030,4580,0800,0000,1219  ;2014 TRAP1: D_D.OR.K[.8000]
U 1219, 0000,003C,6580,0800,1404,7008  ;2015  STATE_K[.10]
U 1008, 0000,003D,0180,0800,0000,1214  ;2016 =0 CALL[SHIFTD]
U 1009, 0000,003C,5980,3C00,0000,121A  ;2017  ID[ACC.2]_D
U 121A, 0000,00FE,0380,0800,0000,0001  ;2018  TRAP.FPA,RETURN1
            ;2019
            ;2020 ;*
            ;2021 ; THE FOLLOWING ROUTINES GENERATE FPA UPC ADDRESSES, AND
            ;2022 ; JUMP TO THE TRAP ROUTINE.
            ;2023 ;-
            ;2024
U 121B, 0818,0038,A580,0800,0000,121C  ;2025 TRP.68: D_K[.60]
U 121C, 0819,0030,0180,0800,0000,1218  ;2026  D_D.OR.K[.8],J/TRAP1
            ;2027
U 121D, 0818,0038,A580,0800,0000,121E  ;2028 TRP.6D: D_K[.60]
U 121E, 0819,0030,8980,0800,0000,1218  ;2029  D_D.OR.K[.D],J/TRAP1
            ;2030
            ;2031
            ;2032
U 121F, 0818,0038,4180,0800,0000,1220  ;2033 TRP.84: D_K[.80]
U 1220, 0819,0014,1180,0800,0000,1218  ;2034  D_D+K[.4],J/TRAP1
            ;2035
U 1221, 0818,0038,4180,0800,0000,1222  ;2036 TRP.87: D_K[.80]
U 1222, 0000,003C,5D80,0800,0000,1223  ;2037  KMX/.7
U 1223, 0819,0014,5D80,0800,0000,1218  ;2038  D_D+K[.7],J/TRAP1
            ;2039
U 1224, 0818,0038,8980,0800,0000,1225  ;2040 D0: D_K[.D]
U 1225, 0100,003C,0180,0800,0000,1226  ;2041  D_D.LEFT2
U 1226, 0100,003E,0180,0800,0000,0001  ;2042  D_D.LEFT2,RETURN1
            ;2043
            ;2044 =0
U 100C, 0000,003D,0180,0800,0000,1224  ;2045 TRP.D1: CALL[D0]
U 100D, 0819,0030,0580,0800,0000,1218  ;2046  D_D.OR.K[.1],J/TRAP1
            ;2047
            ;2048 =0
U 1010, 0000,003D,0180,0800,0000,1227  ;2049 TRP.E1: CALL[E0]
U 1011, 0819,0030,0580,0800,0000,1218  ;2050  D_D.OR.K[.1],J/TRAP1
            ;2051
U 1227, 0818,0038,CD80,0800,0000,1228  ;2052 E0: D_K[.F0]
U 1228, 0819,0026,6580,0800,0000,0001  ;2053  D_D.ANDNOT.K[.10],RETURN1
            ;2054
            ;2055
            ;2056
            ;2057
            ;2058
U 1229, 0818,0038,CD80,0800,0000,122A  ;2059 TRP.F3: D_K[.F0]
U 122A, 0819,0014,0D80,0800,0000,1218  ;2060  D_D+K[.3],J/TRAP1
```

```
            ;2061
            ;2062
U 122B, 0818,0038,CD80,0800,0000,122C  ;2063 TRP.F4: D_K[.F0]
U 122C, 0819,0014,1180,0800,0000,1218  ;2064  D_D+K[.4],J/TRAP1
            ;2065
            ;2066
U 122D, 0818,0038,CD80,0800,0000,122E  ;2067 TRP.F5: D_K[.F0]
U 122E, 0819,0010,1180,0800,0000,1218  ;2068  D_D+K[.4]+1,J/TRAP1
            ;2069
            ;2070
            ;2071
U 122F, 0818,0038,CD80,0800,0000,1230  ;2072 TRP.F8: D_K[.F0]
U 1230, 0819,0014,0180,0800,0000,1218  ;2073  D_D+K[.8],J/TRAP1
            ;2074
U 1231, 0818,0038,CD80,0800,0000,1232  ;2075 TRP.FB: D_K[.F0]
U 1232, 0819,0010,F580,0800,0000,1218  ;2076  D_D+K[.A]+1,J/TRAP1
            ;2077
            ;2078
            ;2079 ;+
            ;2080 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2081 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE
            ;2082 ; IB IS COMPLETELY FULL OF DATA.
            ;2083 ;
            ;2084 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2085 ;-
U 1233, 3000,003C,0180,6000,0000,1234  ;2086 IBLOAD: LOAD.IB,IBC/START
U 1234, 0000,003C,0180,F800,0000,1235  ;2087 SYNC02: MCT/ALLOW.IB.READ
U 1235, 0000,003C,0180,F800,0000,1236  ;2088  MCT/ALLOW.IB.READ
U 1236, 0000,003C,0180,F800,0000,1237  ;2089  MCT/ALLOW.IB READ
U 1237, 0000,003C,0180,F800,0000,1238  ;2090  MCT/ALLOW.IB.READ
U 1238, 0000,003C,0180,F800,0000,1239  ;2091  MCT/ALLOW.IB.READ
U 1239, 1000,003E,0180,F800,0000,0001  ;2092  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2093
            ;2094 ;+
            ;2095 ; THESE ROUTINES WILL LOAD EXPONENT AND SIGN DATA INTO THE SPECIFIED
            ;2096 ; SCRATCH PAD REGISTERS THE DATA IS THEN SHIFTED TO THE APPROPRIATE
            ;2097 ; BIT POSITIONS FOR THE EXPONENT AND SIGN FIELDS.
            ;2098 ;-
            ;2099
            ;2100
            ;2101 ; LA>LB,SA=0,SB=0
            ;2102 ; DATA FOR RE.NEG ASSERTED. (AR>BR,SA NOT EQUAL TO SB)
            ;2103
U 123A, 0018,0038,1180,0AD0,0000,123B  ;2104 LDDATB: R[0A]_K[.4]
U 123B, 0018,0038,0980,0AD8,0000,123C  ;2105  R[0B]_K[.2]
U 123C, 0800,003C,0180,0A50,0000,123D  ;2106  D_R[0A]
U 123D, 0019,0032,4580,0AD0,0000,0001  ;2107  R[0A]_D.OR.K[.8000],RETURN1
            ;2108
            ;2109 ; DATA FOR RE.NEG DEASSERTED. (SA=SB)
            ;2110
U 123E, 0018,0038,1180,0AD0,0000,123F  ;2111 LDDATC:R[0A]_K[.4]  ,
U 123F, 0018,003A,0980,0AD8,0000,0001  ;2112  R[0B]_K[.2],RETURN1
            ;2113
            ;2114
            ;2115 ; LA>LB,SA=0,SB=0
```

```
            ;2116
U 1240, 0818.0038.2980.0800.0000.1012  ;2117 LDDATA: D_K[.34]
U 1012, 0000.003D.5D80.0800.1404.7214  ;2118 =0 STATE_K[.7],CALL[SHIFTD]
U 1013, 0001.003C.0180.0AA8.0000.1241  ;2119  R[5]_D
U 1241, 0818.0038.2D80.0800.0000.1014  ;2120  D_K[.28]
U 1014, 0000.003D.5D80.0800.1404.7214  ;2121 =0 STATE_K[.7],CALL[SHIFTD]
U 1015, 0001.003E.0180.0AB0.0000.0001  ;2122  R[6]_D,RETURN1
            ;2123
            ;2124
            ;2125 ; LA<LB,SA=0,SB=0
            ;2126
U 1242, 0818.0038.2980.0800.0000.1016  ;2127 LDDAT2: D_K[.34]
U 1016, 0000.003D.5D80.0800.1404.7214  ;2128 =0 STATE_K[.7],CALL[SHIFTD]
U 1017, 0001.003C.0180.0AB0.0000.1243  ;2129  R[6]_D
U 1243, 0818.0038.0180.0800.0000.1018  ;2130  D_K[.8]
U 1018, 0000.003D.5D80.0800.1404.7214  ;2131 =0 STATE_K[.7],CALL[SHIFTD]
U 1019, 0001.003E.0180.0AA8.0000.0001  ;2132  R[5]_D,RETURN1
            ;2133
            ;2134
            ;2135 ; LA<LB,SA=0,SB=1
            ;2136
U 1244, 0818.0038.2D80.0800.0000.101A  ;2137 LDDAT3: D_K[.28]
U 101A, 0000.003D.5D80.0800.1404.7214  ;2138 =0 STATE_K[.7],CALL[SHIFTD]
U 101B, 0001.003C.0180.0AA8.0000.1245  ;2139  R[5]_D
U 1245, 0818.0038.2980.0800.0000.101C  ;2140  D_K[.34]
U 101C, 0000.003D.5D80.0800.1404.7214  ;2141 =0 STATE_K[.7],CALL[SHIFTD]
U 101D, 0819.0030.4580.0800.0000.1246  ;2142  D_D.OR.K[.8000]  ; SB=1
U 1246, 0001.003E.0180.0AB0.0000.0001  ;2143  R[6]_D,RETURN1
            ;2144
            ;2145
            ;2146 ; LA=LB,SA=1,SB=0
            ;2147
U 1247, 0818.0038.2980.0800.0000.101E  ;2148 LDDAT5: D_K[.34]
U 101E, 0000.003D.5D80.0800.1404.7214  ;2149 =0 STATE_K[.7],CALL[SHIFTD]
U 101F, 0001.003C.0180.0AB0.0000.1248  ;2150  R[6]_D
U 1248, 0819.0030.4580.0800.0000.1249  ;2151  D_D.OR.K[.8000]  ; SA=1
U 1249, 0001.003E.0180.0AA8.0000.0001  ;2152  R[5]_D,RETURN1
            ;2153
            ;2154 ; LA<LB,SA=1,SB=1
            ;2155
U 124A, 0818.0038.D180.0800.0000.124B  ;2156 LDDAT6: D_K[.C0]
U 124B, 0819.0014.0D80.0800.0000.1020  ;2157  D_D+K[.3]
U 1020, 0000.003D.5D80.0800.1404.7214  ;2158 =0 STATE_K[.7],CALL[SHIFTD]
U 1021, 0819.0030.4580.0800.0000.124C  ;2159  D_D.OR.K[.8000]
U 124C, 0001.003C.0180.0AB0.0000.124D  ;2160  R[6]_D
U 124D, 0818.0038.D180.0800.0000.1022  ;2161  D_K[.C0]
U 1022, 0000.003D.5D80.0800.1404.7214  ;2162 =0 STATE_K[.7],CALL[SHIFTD]
U 1023, 0819.0030.4580.0800.0000.124E  ;2163  D_D.OR.K[.8000]
U 124E, 0001.003E.0180.0AA8.0000.0001  ;2164  R[5]_D,RETURN1
            ;2165
            ;2166
            ;2167 ;LA=LB,SA=0,SB=1
            ;2168
U 124F, 0818.0038.2980.0800.0000.1024  ;2169 LDDAT7: D_K[.34]
U 1024, 0000.003D.5D80.0800.1404.7214  ;2170 =0 STATE_K[.7],CALL[SHIFTD]
```

```
U 1025, 0001,003C,0180,0AA8,0000,1250  ;2171  R[5]_D
U 1250, 0819,0030,4580,0800,0000,1251  ;2172  D_D.OR.K[.8000]
U 1251, 0001,003E,0180,0AB0,0000,0001  ;2173  R[6]_D,RETURN1
            ;2174
            ;2175
            ;2176 ; LA=LB,SA=0,SB=0
            ;2177
U 1252, 0818,0038,2980,0800,0000,1026  ;2178 LDDAT8: D_K[.34]
U 1026, 0000,003D,5D80,0800,1404,7214  ;2179 =0 STATE_K[.7],CALL[SHIFTD]
U 1027, 0001,003C,0180,0AA8,0000,1253  ;2180  R[5]_D
U 1253, 0001,003E,0180,0AB0,0000,0001  ;2181  R[6]_D,RETURN1
            ;2182
            ;2183 ; THIS ROUTINE WILL LOAD THE FAD WITH AR>BR AND ASSERT
            ;2184 ; SA NOT EQUAL TO SB. IT WILL THEN EXECUTE AN ADD ISTRUCTION.
            ;2185 ; THIS WILL GENERATE RESULT NEGATIVE. RESULT NEGATIVE IS
            ;2186 ; (FALU MSB)*(ALU SUB).
            ;2187
            ;2188 =0
U 1028, 2018,0039,7D80,0800,4200,1233  ;2189 RE.NEG: ALU_K[.18],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1029, 0000,003C,0180,0800,0000,102A  ;2190  NOP
U 102A, 0000,003D,0180,0800,0060,1221  ;2191 =0 CALL[TRP.87]  ; LOAD AR,BR,LA,SA
U 102B, 0000,003C,0180,0A50,0000,1254  ;2192  LAB_R[0A]  ; FPA AT 087. LOAD SA,LA
U 1254, 0000,007C,0180,0A50,0000,1255  ;2193  LAB_R[0A],CPSYNC ; GO TO 088
U 1255, 0000,003C,0180,0A58,0000,1256  ;2194  LAB_R[0B]  ; FPA AT 088. LOAD SB,LB
U 1256, 0000,007C,0180,0A58,0000,102C  ;2195  LAB_R[0B],CPSYNC ; GO TO 028
U 102C, 0000,003D,0180,0800,0000,121F  ;2196 =0 CALL[TRP.84]
U 102D, 0000,003C,0180,0800,0000,1257  ;2197  NOP   ; FPA AT 084  (SET RE.NEG)
U 1257, 0000,003C,0180,0800,0000,1258  ;2198  NOP   ; FPA AT 028  (HOLD RE.NEG)
U 1258, 0000,003E,0180,0800,0000,0001  ;2199  RETURN1
            ;2200
            ;2201 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRSSS IN THE D REGISTER AND
            ;2202 ; CONVERT IT TO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2203 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2204
U 1259, 0000,003C,65F8,0800,0084,725A  ;2205 GENTRA: SC_K[.10],Q_0
U 125A, 0D00,003C,8D80,0800,0084,725B  ;2206  D_DAL.SC,SC_K[.1F]
U 125B, 0801,001E,0180,0800,0000,0001  ;2207  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2208
            ;2209 ;+
            ;2210 ; THIS ROUTINE LOADS THE PR IN THE EXPONENT PROCESSOR, FROM R5
            ;2211 ; THE PR IS ROUTED TO THE EALU OUTPUT VIA THE AMX. THE PR PROVIDES
            ;2212 ; A NON-ZERO OUTPUT WHICH PREVENTS THE NORMALIZER MUX FORCING ZEROS
            ;2213 ; ON THE BUS, WHEN THE SIGN PROCESSOR DATA IS READ BACK TO THE CPU.
            ;2214 ;-
            ;2215
            ;2216
U 125C, 0818,0038,E980,0800,0000,125D  ;2217 LD.PR: D_K[.24]
U 125D, 0000,003C,0180,0800,0000,102E  ;2218  NOP
U 102E, 0819,0015,0580,0800,0000,1259  ;2219 =0 D_D+K[.1],CALL[GENTRA]
U 102F, 0000,003C,5980,3C00,0000,125E  ;2220  ID[ACC.2]_D
U 125E, 0000,00FC,0380,0A28,0000,125F  ;2221  TRAP.FPA,LAB_R[5]
U 125F, 0000,007C,0180,0A28,0000,1260  ;2222  LAB_R[5],CPSYNC  ; FPA AT 025. LOAD LA
U 1260, 0000,003C,0180,0800,0000,1261  ;2223  NOP   ; FPA AT 026. LOAD PR
U 1261, 0000,003E,0180,0800,0000,0001  ;2224  RETURN1
            ;2225
```

```
            ;2226 ;
            ;2227 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;2228 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;2229 ;   28: NAD/28   ; NOP
            ;2230 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;2231 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;2232 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;2233 ;  INITIALIZE ITSELF.
            ;2234 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;2235 ;
            ;2236
U 1262, 0818.0038,4580.0800.0000.1263  ;2237 FPINIT: D_K[.8000]
U 1263, 0000.003C,5D80.3C00.0000.1030  ;2238  ID[ACC.CS]_D  ; TURN ON FPA
U 1030, 0818.0039,2D80.0800.0000.1259  ;2239 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1031, 0000.003C,5980.3C00.0000.1264  ;2240  ID[ACC.2]_D
U 1264, 0000.00FC,0380.0800.0000.1032  ;2241  TRAP.FPA  ; TRAP FPA TO 28.
U 1032, 0018.0039,1980.0800.0200.126A  ;2242 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 1033, 0F00.003C,0180.0800.0000.1034  ;2243  D_0
U 1034, 0000.003D,0180.0800.0000.1259  ;2244 =0 CALL,J/GENTRA
U 1035, 0000.003C,5980.3C00.0000.1265  ;2245  ID[ACC.2]_D
U 1265, 0000.00FC,0380.0800.0000.1036  ;2246  TRAP.FPA  ; TRAP FPA TO 0
U 1036, 0818.0039,2D80.0800.0000.1259  ;2247 =0 D_K[.28],CALL,J/GENTRA
U 1037, 0000.003C,5980.3C00.0000.1266  ;2248  ID[ACC.2]_D
U 1266, 0000.00FE,0380.0800.0000.0001  ;2249  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
            ;2250 ;
            ;2251 ;
            ;2252 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2253 ;*0
            ;2254 ;$ 0000.0000, 0000.0000
            ;2255 ;
            ;2256 ;
            ;2257 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2258 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2259 ;
            ;2260
            ;2261 =0
U 1038, 0F00.003D,0180.0800.0000.1259  ;2262 FPIRD: D_0,CALL,J/GENTRA
U 1039, 0000.003C,5980.3C00.0000.1267  ;2263  ID[ACC.2]_D
U 1267, 0000.00FC,0380.0800.0000.1268  ;2264  TRAP.FPA  ; TRAP TO FPA 0
U 1268, 0000.003C,0180.0800.0000.1269  ;2265  NOP
U 1269, 0000.003E,0180.0800.0000.0001  ;2266  RETURN1
            ;2267 ;
            ;2268 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2269 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2270 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2271 ;
            ;2272
U 126A, 2000.003C,0180.0800.0000.126B  ;2273 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 126B, 3000.003C,0180.6000.0000.126C  ;2274  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 126C, 0000.003C,0180.F800.0000.126D  ;2275 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 126D, 0000.003C,0180.F800.0000.126E  ;2276  MCT/ALLOW.IB.READ
U 126E, 0000.003C,0180.F800.0000.126F  ;2277  MCT/ALLOW.IB.READ
U 126F, 0000.003C,0180.F800.0000.1270  ;2278  MCT/ALLOW.IB.READ
U 1270, 0000.003C,0180.F800.0000.1271  ;2279  MCT/ALLOW.IB.READ
U 1271, 1000.003C,0180.F800.0000.1272  ;2280  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
```

```
U 1272, 0000,003E,0180,0800,0000,0001  ;2281  RETURN1
U 1273, 3000,003C,0180,0800,0000,126C  ;2282 WAITIB: IBC/START,J/IBW
            ;2283
            ;2284
            ;2285
            ;2286
            ;2287
            ;2288
            ;2289 ;*
            ;2290 ; THIS ROUTINE LOADS THE FAD AR AND BR LATCHES WITH MULTIPLIER AND
            ;2291 ; MULTIPLICAND OPERANDS. R6,R7,R8,AND R9 MUST BE LOADED IN THE
            ;2292 ; CALLING PROGRAM. THE LA EXPONENT LATCH IS LOADED SO THAT WHEN THE
            ;2293 ; NORMALIZED FRACTION PRODUCTS ARE READ BACK ON BUSA, THE NORMALIZER
            ;2294 ; MUX WILL PASS THE NORMALIZED PRODUCT, AND NOT FORCE ZEROS ON THE BUS.
            ;2295 ;_
            ;2296
            ;2297 =0
U 103A, 0000,003D,0180,0800,0000,121B  ;2298 LD.MUL: CALL[TRP.68]
U 103B, 0018,0038,4180,0AF0,0000,1274  ;2299  R[0E]_K[.80]
U 1274, 0000,003C,0180,0A70,0000,1275  ;2300  LAB_R[0E]  ; FPA AT 068.
U 1275, 0000,007C,0180,0A70,0000,1276  ;2301  LAB_R[0E],CPSYNC ; LOAD LA
U 1276, 0000,007C,0180,0800,0000,1277  ;2302  CPSYNC   ; FPA AT 069.
U 1277, 0000,003C,0180,0A30,0000,1278  ;2303  LAB_R[6]  ; FPA AT 06A.
U 1278, 0000,007C,0180,0A30,0000,1279  ;2304  LAB_R[6],CPSYNC  ; LOAD AR
U 1279, 0000,003C,0180,0A38,0000,127A  ;2305  LAB_R[7]  ; FPA AT 06C.
U 127A, 0000,007C,0180,0A38,0000,127B  ;2306  LAB_R[7],CPSYNC  ; LOAD AR0
U 127B, 0000,003C,0180,0A40,0000,127C  ;2307  LAB_R[8]  ; FPA AT 06D.
U 127C, 0000,007C,0180,0A40,0000,127D  ;2308  LAB_R[8],CPSYNC  ; LOAD BR
U 127D, 0000,003C,0180,0A48,0000,127E  ;2309  LAB_R[9]  ; FPA AT 083.
U 127E, 0000,007C,0180,0A48,0000,127F  ;2310  LAB_R[9],CPSYNC  ; LOAD BR0
U 127F, 0000,003E,0180,0800,0000,0001  ;2311  RETURN1   ; FPA AT 028. LOAD COMPLETE.
            ;2312
            ;2313
            ;2314
            ;2315 ;*
            ;2316 ; THIS ROUTINE LOADS THE IB WITH A MUL FLOAT INSTRUCTION THEN
            ;2317 ; TAKES MULTIPLIER AND MULTIPLICAND OPERANDS PREVIOUSLY LOADED
            ;2318 ; IN THE FAD AR AND BR LATCHES, AND LOADS THE MULTIPLIER OPERAND
            ;2319 ; REGISTERS FOR A MUL FLOAT. THE ROUTINE THEN JUMPS TO SUBROUTINE
            ;2320 ; PROD TO READ BACK THE PRODUCT.
            ;2321
            ;2322 =0
U 103C, 2018,0039,0180,0800,4200,1233  ;2323 MULF: ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ;
U 103D, 0000,003C,0180,0800,0000,103E  ;2324  NOP
U 103E, 0000,003D,0180,0800,0000,1010  ;2325 =0 CALL[TRP.E1]
U 103F, 0000,003C,0180,0800,0000,1280  ;2326  NOP   ; FPA AT 0E1. MC1_AR1
U 1280, 0000,003C,0180,0800,0000,1281  ;2327  NOP   ; FPA AT 0E2. MP1_BR1
U 1281, 0000,003C,0180,0800,0000,1282  ;2328  NOP    ; FPA AT 0C8. START MULTIPLY
U 1282, 0000,003C,0180,0800,0000,1283  ;2329  NOP   ; FPA AT 0C9. CONTINUE MULTIPLY
U 1283, 0000,003C,0180,0800,0000,1284  ;2330  NOP   ; FPA AT 028. MULF R,R DONE.
U 1284, 0000,003C,0180,0800,0000,1285  ;2331  NOP   ; ALLOW PRODUCT TO FINALIZE
U 1285, 0000,003C,0180,0800,0000,1040  ;2332  NOP,J/PROD  ;
            ;2333
            ;2334 ;*
            ;2335 ; THIS ROUTINE TRAPS TO THE FPA UCODE THAT CONTROLS THE TRANSMISSION
```

```
            ;2336 ; OF NORMALIZED AND UNNORMALIZED PRODUCTS BACK TO THE CPU FOR
            ;2337 ; VERIFICATION.
            ;2338 ;-
            ;2339
            ;2340
            ;2341
            ;2342
            ;2343
            ;2344 =0
U 1040, 0000.003D.0180.0800.0000.100C  ;2345 PROD: CALL[TRP.D1]
U 1041, 0000.003C.01D8.0800.0000.1286  ;2346  Q_ACC    ; FPA AT 0D1.
            ;2347      ; SAVE PRODUCT HI AND LO
            ;2348      ; IN AR LATCH AND IN
            ;2349      ; NORMALIZER REGISTER
            ;2350
U 1286, 0001.207C.0180.0988.0000.1287  ;2351  RC[1]_Q,CPSYNC   ; SAVE UNNORMALIZED PRODUCT HI
U 1287, 0000.003C.01D8.0800.0000.1288  ;2352  Q_ACC    ; FPA AT 0D2
U 1288, 0001.207C.0180.0980.0000.1289  ;2353  RC[0]_Q,CPSYNC   ; SAVE UNNORMALIZED PRODUCT LO
U 1289, 0000.003C.01D8.0800.0000.128A  ;2354  Q_ACC    ; FPA AT 0D3.
            ;2355      ; READ NORMALIZED PRODUCT LO
U 128A, 0A00.003C.0180.0800.0000.128B  ;2356  D_ACC    ; FPA AT 0D4,
U 128B, 0000.003E.0180.0800.0000.0001  ;2357  RETURN1    ; READ NORMALIZED PRODUCT HI
            ;2358
            ;2359 =0
U 1042, 0000.003D.0180.0800.0000.1231  ;2360 FAD.CHK: CALL[TRP.FB]
U 1043, 0000.003C.01D8.0800.0000.128C  ;2361  Q_ACC    ; FPA AT 0FB. BUSA_AR1
U 128C, 0001.207C.0180.09F0.0000.128D  ;2362  RC[0E]_Q,CPSYNC   ; SAVE
U 128D, 0A00.003C.0180.0800.0000.128E  ;2363  D_ACC    ; FPA AT 0FC. BUSA_AR0
U 128E, 0001.007C.0180.09E8.0000.128F  ;2364  RC[0D]_D,CPSYNC   ; SAVE
U 128F, 0000.003C.01D8.0800.0000.1290  ;2365  Q_ACC    ; FPA AT 0FD. BUSA_BR1
U 1296, 0001.207C.0180.09E0.0000.1291  ;2366  RC[0C]_Q,CPSYNC   ; SAVE
U 1291, 0A00.003C.0180.0800.0000.1292  ;2367  D_ACC    ; FPA AT 0FE. BUSA_BR0
U 1292, 0001.003C.0180.09D8.0000.1293  ;2368  RC[0B]_D   ; SAVE
            ;2369
            ;2370
            ;2371 ;*
            ;2372 ; THESE SUBROUTINES LOAD EXPONENT AND SIGN DATA INTO THE SPECIFIED
            ;2373 ; SCRATCH PAD REGISTERS.
            ;2374 ;-
            ;2375
            ;2376
            ;2377 ; LA=LB.  SA_1, SB_0.
            ;2378
U 1293, 0818.0038.29F8.0800.0000.1294  ;2379 EXP.D0: D_K[.34],Q_0
U 1294, 0000.003C.5D80.0800.0084.7295  ;2380  SC_K[.7]
U 1295, 0D00.003C.0180.0800.0000.1296  ;2381  D_DAL.SC
U 1296, 0001.003C.0180.0AB0.0000.1297  ;2382  R[6]_D    ; LB_34,SB_0
U 1297, 0019.0030.4580.0AA8.0000.1298  ;2383  ALU_D.OR.K[.8000],R[5]_ALU ; LA_34,SA_1
U 1298, 0000.003E.0180.0800.0000.0001  ;2384  RETURN1
            ;2385
            ;2386
            ;2387 ; LA=LB+1. SA_1, SB_0.
            ;2388
U 1299, 0818.0038.29F8.0800.0000.129A  ;2389 EXP.D1: D_K[.34],Q_0
U 129A, 0000.003C.5D80.0800.0084.729B  ;2390  SC_K[.7]
```

```
U 129B, 0D00,003C,0180,0800,0000,129C  ;2391  D_DAL.SC
U 129C, 0001,003C,0180,0AB0,0000,129D  ;2392  R[6]_D    ; LB_34,SB_0
U 129D, 0819,0030,4180,0800,0000,129E  ;2393  D_D.OR.K[.80]
U 129E, 0019,0030,4580,0AA8,0000,129F  ;2394  ALU_D.OR.K[.8000],R[5]_ALU ; LA_35,SA_1
U 129F, 0000,003E,0180,0800,0000,0001  ;2395  RETURN1
           ;2396
           ;2397 ; LB=LA+1. SA_1, SB_0.
           ;2398
U 12A0, 0818,0038,29F8,0800,0000,12A1  ;2399 EXP.D2: D_K[.34],Q_0
U 12A1, 0000,003C,5D80,0800,0084,72A2  ;2400  SC_K[.7]
U 12A2, 0D00,003C,0180,0800,0000,12A3  ;2401  D_DAL.SC
U 12A3, 0019,0030,4180,0AB0,0000,12A4  ;2402  ALU_D.OR.K[.80],R[6]_ALU ; LB_35,SB_0
U 12A4, 0019,0030,4580,0AA8,0000,12A5  ;2403  ALU_D.OR.K[.8000],R[5]_ALU ; LA_34,SA_1
U 12A5, 0000,003E,0180,0800,0000,0001  ;2404  RETURN1
           ;2405
           ;2406 ; LA=LB+1. SA_1, SB_1.
           ;2407
U 12A6, 0818,0038,29F8,0800,0000,12A7  ;2408 EXP.D3: D_K[.34],Q_0
U 12A7, 0000,003C,5D80,0800,0084,72A8  ;2409  SC_K[.7]
U 12A8, 0D00,003C,0180,0800,0000,12A9  ;2410  D_DAL.SC
U 12A9, 0019,0030,4580,0AB0,0000,12AB  ;2411  ALU_D.OR.K[.8000],R[6]_ALU ; LB_34,SB_1
U 12AB, 0819,0030,4180,0800,0000,12AC  ;2412  D_D.OR.K[.80]
U 12AC, 0019,0030,4580,0AA8,0000,12AD  ;2413  ALU_D.OR.K[.8000],R[5]_ALU ; LA_35,SA_1
U 12AD, 0000,003E,0180,0800,0000,0001  ;2414  RETURN1
           ;2415
           ;2416 ; LA=LB.  SA_0, SB_1.
           ;2417
U 12AE, 0818,0038,29F8,0800,0000,12AF  ;2418 EXP.D4: D_K[.34],Q_0
U 12AF, 0000,003C,5D80,0800,0084,72B0  ;2419  SC_K[.7]
U 12B0, 0D00,003C,0180,0800,0000,12B1  ;2420  D_DAL.SC
U 12B1, 0001,003C,0180,0AA8,0000,12B2  ;2421  R[5]_D    ; LA_34,SA_0
U 12B2, 0019,0030,4580,0AB0,0000,12B3  ;2422  ALU_D.OR.K[.8000],R[6]_ALU ; LB_34,SB_1
U 12B3, 0000,003E,0180,0800,0000,0001  ;2423  RETURN1
           ;2424
           ;2425
           ;2426 ;*
           ;2427 ; THIS SUBROUTINE  LOADS THE FAD AR AND BR LATCHES
           ;2428 ; WITH MULTIPLIER OPERANDS THAT PRODUCE THE PRODUCT
           ;2429 ; (0.01110..). THE MULTIPLIER PRODUCT IS THEN LOADED
           ;2430 ; INTO THE FAD AR LATCH. AT THE SAME TIME THE FPA
           ;2431 ; BUS A BIT 32 IS USED TO LATCH THE A OUT OF NORM
           ;2432 ; CONDITION ON THE FCT.
           ;2433 ;-
           ;2434
U 12B4, 0018,0038,CD80,0AB0,0000,12B5  ;2435 AONORM: R[6]_K[.F0]  ; AR1_0.1110...
U 12B5, 0003,003C,0180,0AB8,0000,12B6  ;2436  R[7]_0    ; AR0_0
U 12B6, 0003,003C,0180,0AC0,0000,12B7  ;2437  R[8]_0    ; BR_0.1000...
U 12B7, 0003,003C,0180,0AC8,0000,1044  ;2438  R[9]_0    ; BR0_0
U 1044, 0000,003D,0180,0800,0000,103A  ;2439 =0 CALL[LD.MUL]  ; AR_0.1110.. ,BR_0.1000..
U 1045, 0000,003C,0180,0800,0000,1046  ;2440  NOP
U 1046, 0000,003D,0180,0800,0000,103C  ;2441 =0 CALL[MULF]  ; MP1_0.1110... ,MC1_0.1000...
U 1047, 0000,003E,0180,0800,0000,0001  ;2442  RETURN1   ; LATCH A OUT OF NORM.
           ;2443
           ;2444 ;*
           ;2445 ; THIS SUBROUTINE LOADS BR1,BR0 WITH THE CONTENTS
```

```
            ;2446 ; OF R8, R9.
            ;2447 ;-
            ;2448
            ;2449 =0
U 1048, 0018,0038,4980,0AC0,0000,1049  ;2450 LD.BR: R[8]_K[.FF]
U 1049, 0003,003C,0180,0AC8,0000,104A  ;2451  R[9]_0
U 104A, 0000,003D,0180,0800,0000,121D  ;2452 =0 CALL[TRP.6D]
U 104B, 0000,003C,0180,0A40,0000,12B8  ;2453  LAB_R[8]  ; FPA AT 06D
U 12B8, 0000,007C,0180,0A40,0000,12B9  ;2454  LAB_R[8],CPSYNC  ; LOAD BR
U 12B9, 0000,003C,0180,0A48,0000,12BA  ;2455  LAB_R[9]  ; FPA AT 083
U 12BA, 0000,007C,0180,0A48,0000,12BB  ;2456  LAB_R[9],CPSYNC  ; LOAD BR0.
U 12BB, 0000,003E,0180,0800,0000,0001  ;2457  RETURN1  ; FPA AT 028. LOAD COMPLETE.
            ;2458
            ;2459 ; THESE LOCATIONS ARE RESERVED FOR THE WCS DEBUGGER ROUTINE.
            ;2460
U 1120, 0000,003C,0180,0800,0000,1121  ;2461 1120: NOP
U 1121, 0000,003C,0180,0800,0000,1122  ;2462 1121: NOP
U 1122, 0000,003C,0180,0800,0000,1123  ;2463 1122: NOP
U 1123, 0000,003C,0180,0800,0000,1124  ;2464 1123: NOP
U 1124, 0000,003C,0180,0800,0000,1125  ;2465 1124: NOP
U 1125, 0000,003C,0180,0800,0000,1126  ;2466 1125: NOP
U 1126, 0000,003C,0180,0800,0000,1127  ;2467 1126: NOP
U 1127, 0000,003C,0180,0800,0000,1128  ;2468 1127: NOP
U 1128, 0000,003C,0180,0800,0000,1129  ;2469 1128: NOP
U 1129, 0000,003C,0180,0800,0000,112A  ;2470 1129: NOP
U 112A, 0000,003C,0180,0800,0000,112B  ;2471 112A: NOP
U 112B, 0000,003C,0180,0800,0000,112C  ;2472 112B: NOP
U 112C, 0000,003C,0180,0800,0000,112D  ;2473 112C: NOP
U 112D, 0000,003C,0180,0800,0000,112E  ;2474 112D: NOP
U 112E, 0000,003C,0180,0800,0000,112F  ;2475 112E: NOP
U 112F, 0000,003C,0180,0800,0000,104C  ;2476 112F: NOP
            ;2477
```

```
;2478  .PAGE "TEST 281 SIGN LOGIC FOR ADD/SUB WITH LA=LB"
;2479 ;*****************************************************************
;2480 ;**
;2481 ; TEST SIGN LOGIC FOR ADD/SUB WITH LA=LB
;2482 ;
;2483 ; TEST DESCRIPTION
;2484 ;
;2485 ; THE SIGN DECISION LOGIC FOR ADD AND SUBTRACT WITH
;2486 ; LA=LB, IS CHECKED IN THIS TEST. THE RESULTING SIGN
;2487 ; OUTPUT IS  CHECKED  WITH  ONLY THE  REQUIRED INPUT
;2488 ; AND SIGN MUX SELECT TO THE SIGN OUT PUT MUX ASSERTED.
;2489 ; THEN ALL POSSIBLE COMBINATIONS OF SIGN INPUT AND MUX
;2490 ; SELECT ARE ASSERTED, AND THE SIGN OUT PUT VERIFIED.
;2491 ; POLYADDL IS VERIFIED NOT STUCK AT (0).
;2492 ;
;2493 ; THE RESULT NEGATIVE INPUT IS GENERATED BY LOADING
;2494 ; THE FAD AR AND BR LATCHES WITH AR>BR, ASSERTING
;2495 ; AN ADD OPCODE, AND MAKING THE OPERAND SIGNS
;2496 ; DIFFERENT.
;2497 ;
;2498 ; FPA UCODE USED
;2499 ; LOCATION CONTENTS
;2500 ; 082: NRC/LD,FADC/A.B,NAD/PSTALL
;2501 ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
;2502 ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,NAD/0F6
;2503 ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
;2504 ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/27
;2505 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2506 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/0CA
;2507 ; 0CA: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2508 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2509 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2510 ; 083: BSC/R,SCR.CPU,FADCBR0,WAIT,NAD/028
;2511 ; 0E1: FADC/A,BSC/FAL.HL,OPLD/MC1,NAD/0E2
;2512 ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;2513 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2514 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2515 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D4
;2516 ; 0D2: FADC/A,BSC/NL,BSC/FAL.LH,WAIT,NAD/0D3
;2517 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2518 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2519 ;
;2520 ;
;2521 ; LOGIC DESCRIPTION
;2522 ;
;2523 ; THE FAD MSB AND THE FCT ALU SUBTRACT SIGNALS ARE
;2524 ; COMBINED AND LATCHED ON THE FNM MODULE. RES.NEG H
;2525 ; IS THEN BROUGHT OVER TO THE FCT. THE REST OF THE
;2526 ; LOGIC IS ON THE FCT.
;2527 ;
;2528 ; ERROR DESCRIPTION
;2529 ; EXPECTED SIGN RESULT
;2530 ; RECEIVED SIGN RESULT
;2531 ;
;2532 ; SYNC POINT DESCRIPTION
```

```
                ;2533 ;
                ;2534 ;--
                ;2535 ;*************************************************************************
                ;2536 ;/////////////////////////////////////////
                ;2537 ;*
                ;2538 ; SUBTEST 1
                ;2539 ; (RES.NEG).XOR.(-SB(1)H)*(A=B)*[-(POLYADDL)*(-IR2H)]*(-A OUT OF NORM L)
                ;2540 ;-
                ;2541 =0
U 104C. 0018.0039.0980.09F8.0000.11F8   ;2542 SGN5: NEWTST[.2]
U 104D. 0000.003C.0180.0800.0000.1050   ;2543  NOP
U 1050. 0000.003D.0180.0800.0000.120F   ;2544 =0 SUBTEST
U 1051. 0000.003C.0180.0800.0000.1052   ;2545  NOP
U 1052. 0000.003D.0180.0800.0000.123A   ;2546 =0 CALL[LDDATB]  ; DATA FOR ASSERTING RE.NEG
U 1053. 0000.003C.0180.0800.0000.1054   ;2547  NOP
U 1054. 0000.003D.0180.0800.0000.1028   ;2548 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 1055. 0018.0038.4580.09F0.0000.1056   ;2549  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 1056. 0000.003D.0180.0800.0000.1247   ;2550 =0 CALL[LDDAT5]  ; LA=LB,SA=1,SB=0
U 1057. 0000.003C.0180.0800.0000.1058   ;2551  NOP
U 1058. 0000.003D.0180.0800.0000.125C   ;2552 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 1059. 0000.003C.0180.0800.0000.105C   ;2553  NOP
U 105C. 2018.0039.0180.0800.4200.1233   ;2554 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 105D. 0000.003C.0180.0800.0000.1060   ;2555  NOP
U 1060. 0000.003D.0180.0800.0000.122F   ;2556 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1061. 0000.007C.0180.0800.0000.12BC   ;2557  CPSYNC
U 12BC. 0000.003C.01D8.0800.0000.12BD   ;2558  Q_ACC
U 12BD. 0019.2034.45C0.0800.0000.12BE   ;2559  Q_Q.AND.K[.8000]
U 12BE. 0001.203C.0180.09E8.0000.12BF   ;2560  RC[0D]_Q
U 12BF. 0011.2020.0180.0970.0010.12C0   ;2561  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12C0. 0000.013C.0180.0800.0000.1062   ;2562  Z?
U 1062. 0000.003D.0180.0800.0000.120B   ;2563 =0 ERROR1
U 1063. 0000.003C.0180.0800.0000.1064   ;2564  NOP
                ;2565 ;/////////////////////////////////////////
                ;2566 ;*
                ;2567 ; SUBTEST 2
                ;2568 ; (RES.NEG).XOR.(SB(1)H)*(A=B)*[-(POLYADDL)*(-IR2H)]*(-A OUT OF NORM L)
                ;2569 ;-
U 1064. 0000.003D.0180.0800.0000.120F   ;2570 =0 SUBTEST
U 1065. 0000.003C.0180.0800.0000.1066   ;2571  NOP
U 1066. 0000.003D.0180.0800.0000.120A   ;2572 =0 ERLOOP
U 1067. 0000.003C.0180.0800.0000.1068   ;2573  NOP
U 1068. 0000.003D.0180.0800.0000.123A   ;2574 =0 CALL[LDDATB]  ; DATA FOR ASSERTING RE.NEG
U 1069. 0000.003C.0180.0800.0000.106A   ;2575  NOP
U 106A. 0000.003D.0180.0800.0000.1028   ;2576 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 106B. 0003.003C.0180.09F0.0000.106C   ;2577  RC[0E]_0  ; EXPECTED SIGN RESULT
U 106C. 0000.003D.0180.0800.0000.124F   ;2578 =0 CALL[LDDAT7]  ; LA=LB,SA=0,SB=1
U 106D. 0000.003C.0180.0800.0000.106E   ;2579  NOP
U 106E. 0000.003D.0180.0800.0000.125C   ;2580 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 106F. 0000.003C.0180.0800.0000.1070   ;2581  NOP
U 1070. 2018.0039.0180.0800.4200.1233   ;2582 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 1071. 0000.003C.0180.0800.0000.1072   ;2583  NOP
U 1072. 0000.003D.0180.0800.0000.122F   ;2584 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1073. 0000.007C.0180.0800.0000.12C1   ;2585  CPSYNC
U 12C1. 0000.003C.01D8.0800.0000.12C2   ;2586  Q_ACC
U 12C2. 0019.2034.45C0.0800.0000.12C3   ;2587  Q_Q.AND.K[.8000]
```

```
U 12C3. 0001.203C.0180.09E8.0000.12C4  ;2588  RC[0D]_Q
U 12C4. 0011.2020.0180.0970.0010.12C5  ;2589  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12C5. 0000.013C.0180.0800.0000.1074  ;2590  Z?
U 1074. 0000.003D.0180.0800.0000.120B  ;2591 =0 ERROR1
U 1075. 0000.003C.0180.0800.0000.1076  ;2592  NOP
          ;2593 ;/////////////////////////////////////////
          ;2594 ;*
          ;2595 ; SUBTEST 3
          ;2596 ; (RES.NEG).XOR.(-SB(1)H)*(A>B)*[(-POLYADDL)*(IR2H)]*(-A OUT OF NORM L)
          ;2597 ;-
U 1076. 0000.003D.0180.0800.0000.120F  ;2598 =0 SUBTEST
U 1077. 0000.003C.0180.0800.0000.1078  ;2599  NOP
U 1078. 0000.003D.0180.0800.0000.120A  ;2600 =0 ERLOOP
U 1079. 0000.003C.0180.0800.0000.107A  ;2601  NOP
U 107A. 0000.003D.0180.0800.0000.123A  ;2602 =0 CALL[LDDATB]  ; DATA FOR ASSERTING RE.NEG
U 107B. 0000.003C.0180.0800.0000.107C  ;2603  NOP
U 107C. 0000.003D.0180.0800.0000.1028  ;2604 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 107D. 0003.003C.0180.09F0.0000.107E  ;2605  RC[0E]_0  ; EXPECTED SIGN RESULT
U 107E. 0000.003D.0180.0800.0000.1240  ;2606 =0 CALL[LDDATA]  ; LA>LB,SB=0,SA=0
U 107F. 0000.003C.0180.0800.0000.1080  ;2607  NOP
U 1080. 0000.003D.0180.0800.0000.125C  ;2608 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 1081. 0000.003C.0180.0800.0000.1082  ;2609  NOP
U 1082. 2018.0039.0180.0800.4200.1233  ;2610 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 1083. 0000.003C.0180.0800.0000.1084  ;2611  NOP
U 1084. 0000.003D.0180.0800.0000.122F  ;2612 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1085. 0000.007C.0180.0800.0000.12C6  ;2613  CPSYNC
U 12C6. 0000.003C.01D8.0800.0000.12C7  ;2614  Q_ACC
U 12C7. 0019.2034.45C0.0800.0000.12C8  ;2615  Q_Q.AND.K[.8000]
U 12C8. 0001.203C.0180.09E8.0000.12C9  ;2616  RC[0D]_Q
U 12C9. 0011.2020.0180.0970.0010.12CA  ;2617  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12CA. 0000.013C.0180.0800.0000.1086  ;2618  Z?
U 1086. 0000.003D.0180.0800.0000.120B  ;2619 =0 ERROR1
U 1087. 0000.003C.0180.0800.0000.1088  ;2620  NOP
          ;2621 ;/////////////////////////////////////////
          ;2622 ;*
          ;2623 ; SUBTEST 4
          ;2624 ; (-RES.NEG).XOR.(-SB(1)H)*(A=B)*[-(POLYADDL)*(-IR2H)]*(-A OUT OF NORM L)
          ;2625 ;-
U 1088. 0000.003D.0180.0800.0000.120F  ;2626 =0 SUBTEST
U 1089. 0000.003C.0180.0800.0000.108A  ;2627  NOP
U 108A. 0000.003D.0180.0800.0000.120A  ;2628 =0 ERLOOP
U 108B. 0000.003C.0180.0800.0000.108C  ;2629  NOP
U 108C. 0000.003D.0180.0800.0000.123E  ;2630 =0 CALL[LDDATC]  ; DATA FOR DEASSERTING RE.NEG
U 108D. 0000.003C.0180.0800.0000.108E  ;2631  NOP
U 108E. 0000.003D.0180.0800.0000.1028  ;2632 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 108F. 0003.003C.0180.09F0.0000.1090  ;2633  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1090. 0000.003D.0180.0800.0000.1252  ;2634 =0 CALL[LDDAT8]  ; LA=LB,SA=0,SB=0
U 1091. 0000.003C.0180.0800.0000.1092  ;2635  NOP
U 1092. 0000.003D.0180.0800.0000.125C  ;2636 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 1093. 0000.003C.0180.0800.0000.1094  ;2637  NOP
U 1094. 2018.0039.6580.0800.4200.1233  ;2638 =0 ALU_K[.10],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 1095. 0000.003C.0180.0800.0000.1096  ;2639  NOP
U 1096. 0000.003D.0180.0800.0000.122F  ;2640 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1097. 0000.007C.0180.0800.0000.12CB  ;2641  CPSYNC
U 12CB. 0000.003C.01D8.0800.0000.12CC  ;2642  Q_ACC
```

```
U 12CC. 0019,2034,45C0,0800,0000,12CD  ;2643  Q_Q.AND.K[.8000]
U 12CD. 0001,203C,0180,09E8,0000,12CE  ;2644  RC[0D]_Q
U 12CE. 0011,2020,0180,0970,0010,12CF  ;2645  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12CF. 0000,013C,0180,0800,0000,1098  ;2646  Z?
U 1098. 0000,003D,0180,0800,0000,120B  ;2647 =0 ERROR1
U 1099. 0000,003C,0180,0800,0000,109A  ;2648  NOP
          ;2649 ;/////////////////////////////////////////
          ;2650 ;*
          ;2651 ; SUBTEST 5
          ;2652 ; (-RES.NEG).XOR.SB(1)H)*(A=B)*[-(POLYADDL*(-IR2H)]*(-A OUT OF NORM L)
          ;2653 ;-
U 109A. 0000,003D,0180,0800,0000,120F  ;2654 =0 SUBTEST
U 109B. 0000,003C,0180,0800,0000,109C  ;2655  NOP
U 109C. 0000,003D,0180,0800,0000,120A  ;2656 =0 ERLOOP
U 109D. 0000,003C,0180,0800,0000,109E  ;2657  NOP
U 109E. 0000,003D,0180,0800,0000,123E  ;2658 =0 CALL[LDDATC]  ; DATA FOR DEASSERTING RE.NEG
U 109F. 0000,003C,0180,0800,0000,10A0  ;2659  NOP
U 10A0. 0000,003D,0180,0800,0000,1028  ;2660 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 10A1. 0018,0038,4580,09F0,0000,10A2  ;2661  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10A2. 0000,003D,0180,0800,0000,1247  ;2662 =0 CALL[LDDAT5]  ; LA=LB,SA=1,SB=0
U 10A3. 0800,003C,0180,0A30,0000,12D0  ;2663  D_R[6]
U 12D0. 0019,0030,4580,0AB0,0000,10A4  ;2664  R[6]_D.OR.K[.8000] ; SB=1
U 10A4. 0000,003D,0180,0800,0000,125C  ;2665 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 10A5. 0000,003C,0180,0800,0000,10A6  ;2666  NOP
U 10A6. 2018,0039,0180,0800,4200,1233  ;2667 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 10A7. 0000,003C,0180,0800,0000,10A8  ;2668  NOP
U 10A8. 0000,003D,0180,0800,0000,122F  ;2669 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10A9. 0000,007C,0180,0800,0000,12D1  ;2670  CPSYNC
U 12D1. 0000,003C,01D8,0800,0000,12D2  ;2671  Q_ACC
U 12D2. 0019,2034,45C0,0800,0000,12D3  ;2672  Q_Q.AND.K[.8000]
U 12D3. 0001,203C,0180,09E8,0000,12D4  ;2673  RC[0D]_Q
U 12D4. 0011,2020,0180,0970,0010,12D5  ;2674  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12D5. 0000,013C,0180,0800,0000,10AA  ;2675  Z?
U 10AA. 0000,003D,0180,0800,0000,120B  ;2676 =0 ERROR1
U 10AB. 0000,003C,0180,0800,0000,10AC  ;2677  NOP
          ;2678 ;/////////////////////////////////////////
          ;2679 ;*
          ;2680 ; SUBTEST 6
          ;2681 ; (RES.NEG).XOR.(-SB(1))*(A=B)*[-(POLYADDL)*(IR2H)]*(-A OUT OF NORM L)
          ;2682 ;-
U 10AC. 0000,003D,0180,0800,0000,120F  ;2683 =0 SUBTEST
U 10AD. 0000,003C,0180,0800,0000,10AE  ;2684  NOP
U 10AE. 0000,003D,0180,0800,0000,120A  ;2685 =0 ERLOOP
U 10AF. 0000,003C,0180,0800,0000,10B0  ;2686  NOP
U 10B0. 0000,003D,0180,0800,0000,123A  ;2687 =0 CALL[LDDATB]  ; DATA FOR ASSERTING RE.NEG
U 10B1. 0000,003C,0180,0800,0000,10B2  ;2688  NOP
U 10B2. 0000,003D,0180,0800,0000,1028  ;2689 =0 CALL[RE.NEG]  ; (FALU MSB*ALU SUB)
U 10B3. 0003,003C,0180,09F0,0000,10B4  ;2690  RC[0E]_0  ; EXPECTED SIGN RESULT
U 10B4. 0000,003D,0180,0800,0000,1252  ;2691 =0 CALL[LDDAT8]  ; LA=LB,SA=0,SB=0
U 10B5. 0000,003C,0180,0800,0000,10B6  ;2692  NOP
U 10B6. 0000,003D,0180,0800,0000,125C  ;2693 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 10B7. 0000,003C,0180,0800,0000,10B8  ;2694  NOP
U 10B8. 2018,0039,2180,0800,4200,1233  ;2695 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS MULF R,R
U 10B9. 0000,003C,0180,0800,0000,10BA  ;2696  NOP
U 10BA. 0000,003D,0180,0800,0000,122F  ;2697 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
```

```
U 10BB, 0000,007C,0180,0800,0000,12D6  ;2698  CPSYNC
U 12D6, 0000,003C,01D8,0800,0000,12D7  ;2699  Q_ACC
U 12D7, 0019,2034,45C0,0800,0000,12D8  ;2700  Q_Q.AND.K[.8000]
U 12D8, 0001,203C,0180,09E8,0000,12D9  ;2701  RC[0D]_Q
U 12D9, 0011,2020,0180,0970,0010,12DA  ;2702  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12DA, 0000,013C,0180,0800,0000,10BC  ;2703  Z?
U 10BC, 0000,003D,0180,0800,0000,120B  ;2704 =0 ERROR1
U 10BD, 0000,003C,0180,0800,0000,10BE  ;2705  NOP
            ;2706 ;/////////////////////////////////////////
            ;2707 ;*
            ;2708 ; SUBTEST 7
            ;2709 ; (RES.NEG).XOR.(-SB(1)H)*(A>B)*[POLYADDL)+(IR2H)]*(A OUT OF NORM L)
            ;2710 ; ALU SUBTRACT AND EXPONENT DIFFERENCE IS EQUAL TO (1) LA=LB+1.
            ;2711 ; EXPECTED SIGN RESULT LO.
            ;2712 ;-
U 10BE, 0000,003D,0180,0800,0000,120F  ;2713 =0 SUBTEST
U 10BF, 0000,003C,0180,0800,0000,10C0  ;2714  NOP
U 10C0, 0000,003D,0180,0800,0000,120A  ;2715 =0 ERLOOP
U 10C1, 0003,003C,0180,09F0,0000,10C2  ;2716  RC[0E]_0   ; EXPECTED SIGN RESULT
U 10C2, 0000,003D,0180,0800,0000,12B4  ;2717 =0 CALL[AONORM]  ; ASSERT A OUT OF NORM
U 10C3, 0000,003C,0180,0800,0000,10C4  ;2718  NOP   ; LATCH AON ON THE FCT.
U 10C4, 0000,003D,0180,0800,0000,1048  ;2719 =0 CALL[LD.BR]  ; BR1_0.1111111..
U 10C5, 0000,003C,0180,0800,0000,10C6  ;2720  NOP   ; BR0_0
U 10C6, 0000,003D,0180,0800,0000,1299  ;2721 =0 CALL[EXP.D1]  ; LA_35,SA_1
U 10C7, 0000,003C,0180,0800,0000,10C8  ;2722  NOP   ; LB_34,SB_0
U 10C8, 0000,003D,0180,0800,0000,125C  ;2723 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 10C9, 0000,003C,0180,0800,0000,10CA  ;2724  NOP
U 10CA, 2018,0039,0180,0800,4200,1233  ;2725 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 10CB, 0000,003C,0180,0800,0000,10CC  ;2726  NOP
U 10CC, 0000,003D,0180,0800,0000,122F  ;2727 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10CD, 0000,003C,0180,0800,0000,10CE  ;2728  NOP   ; GENERATE RES. NEG.
U 10CE, 0000,003D,0180,0800,0000,121F  ;2729 =0 CALL[TRP.84]
U 10CF, 0000,003C,0180,0800,0000,12DB  ;2730  NOP   ; FPA AT 084 (SET RES. NEG.)
U 12DB, 0000,003C,0180,0800,0000,10D0  ;2731  NOP   ; FPA AT 028 (HOLD RES. NEG.)
U 10D0, 0000,003D,0180,0800,0000,122F  ;2732 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10D1, 0000,007C,0180,0800,0000,12DC  ;2733  CPSYNC   ; GENERATE FINAL SIGN RESULT
U 12DC, 0000,003C,01D8,0800,0000,12DD  ;2734  Q_ACC
U 12DD, 0019,2034,45C0,0800,0000,12DE  ;2735  Q_Q.AND.K[.8000]
U 12DE, 0001,203C,0180,09E8,0000,12DF  ;2736  RC[0D]_Q
U 12DF, 0011,2020,0180,0970,0010,12E0  ;2737  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12E0, 0000,013C,0180,0800,0000,10D2  ;2738  Z?
U 10D2, 0000,003D,0180,0800,0000,120D  ;2739 =0 ERROR2
U 10D3, 0000,003C,0180,0800,0000,10D4  ;2740  NOP
            ;2741 ;/////////////////////////////////////////
            ;2742 ;*
            ;2743 ; SUBTEST 8
            ;2744 ; (RES.NEG).XOR.(SB(1)H)*(A>B)*[POLYADDL)+(IR2H)]*(A OUT OF NORM L)
            ;2745 ; ALU SUBTRACT AND EXPONENT DIFFERENCE IS EQUAL TO (1) LA=LB+1.
            ;2746 ; EXPECTED SIGN RESULT HI.
            ;2747 ;-
            ;2748
            ;2749
U 10D4, 0000,003D,0180,0800,0000,120F  ;2750 =0 SUBTEST
U 10D5, 0000,003C,0180,0800,0000,10D6  ;2751  NOP
U 10D6, 0000,003D,0180,0800,0000,120A  ;2752 =0 ERLOOP
```

```
U 10D7. 0018.0038.4580.09F0.0000.10D8  ;2753  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10D8. 0000.003D.0180.0800.0000.12B4  ;2754 =0 CALL[AONORM]  ; ASSERT A OUT OF NORM
U 10D9. 0000.003C.0180.0800.0000.10DA  ;2755  NOP   ; LATCH AON ON THE FCT.
U 10DA. 0000.003D.0180.0800.0000.1048  ;2756 =0 CALL[LD.BR]  ; BR1_0.1111111..
U 10DB. 0000.003C.0180.0800.0000.10DC  ;2757  NOP   ; BR0_0
U 10DC. 0000.003D.0180.0800.0000.12A6  ;2758 =0 CALL[EXP.D3]  ; LA_35,SA_1
U 10DD. 0000.003C.0180.0800.0000.10DE  ;2759  NOP   ; LB_34,SB_1
U 10DE. 0000.003D.0180.0800.0000.125C  ;2760 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 10DF. 0000.003C.0180.0800.0000.10E0  ;2761  NOP
U 10E0. 2018.0039.8580.0800.4200.1233  ;2762 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS SUB R,R
U 10E1. 0000.003C.0180.0800.0000.10E2  ;2763  NOP
U 10E2. 0000.003D.0180.0800.0000.122F  ;2764 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10E3. 0000.003C.0180.0800.0000.10E4  ;2765  NOP   ; GENERATE RES. NEG.
U 10E4. 0000.003D.0180.0800.0000.121F  ;2766 =0 CALL[TRP.84]
U 10E5. 0000.003C.0180.0800.0000.12E1  ;2767  NOP   ; FPA AT 084 (SET RES. NEG.)
U 12E1. 0000.003C.0180.0800.0000.10E6  ;2768  NOP   ; FPA AT 028 (HOLD RES. NEG.)
U 10E6. 0000.003D.0180.0800.0000.122F  ;2769 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10E7. 0000.007C.0180.0800.0000.12E2  ;2770  CPSYNC   ; GENERATE FINAL SIGN RESULT
U 12E2. 0000.003C.01D8.0800.0000.12E3  ;2771  Q_ACC
U 12E3. 0019.2034.45C0.0800.0000.12E4  ;2772  Q_Q.AND.K[.8000]
U 12E4. 0001.203C.0180.09E8.0000.12E5  ;2773  RC[0D]_Q
U 12E5. 0011.2020.0180.0970.0010.12E6  ;2774  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12E6. 0000.013C.0180.0800.0000.10E8  ;2775  Z?
U 10E8. 0000.003D.0180.0800.0000.120B  ;2776 =0 ERROR1
U 10E9. 0000.003C.0180.0800.0000.10EA  ;2777  NOP
        ;2778
```

```
;2779  .PAGE "TEST 282 SIGN LOGIC FOR ADD/SUB WITH LA<LB"
;2780  ;*****************************************************************
;2781  ;**
;2782  ; TEST SIGN LOGIC FOR ADD/SUB WITH LA<LB
;2783  ;
;2784  ; TEST DESCRIPTION
;2785  ;
;2786  ; THE SIGN DECISION LOGIC FOR ADD AND SUBTRACT WITH
;2787  ; LA<LB IS EXERCISED IN THIS TEST. THE RESULTING SIGN
;2788  ; OUTPUT IS CHECKED WITH  ONLY THE  REQUIRED INPUT
;2789  ; AND SIGN MUX SELECT TO THE SIGN OUT PUT MUX ASSERTED.
;2790  ; THEN ALL POSSIBLE COMBINATIONS OF SIGN INPUT AND MUX
;2791  ; SELECT ARE ASSERTED, AND THE SIGN OUT PUT VERIFIED.
;2792  ;
;2793  ;
;2794  ; FPA UCODE USED
;2795  ; LOCATION CONTENTS
;2796  ; 0F6: BSC/NH,SGNCFALHL.RES,NAD/PSTALL
;2797  ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,NAD/0F6
;2798  ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
;2799  ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/27
;2800  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2801  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD.0CA
;2802  ; 0CA: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2803  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2804  ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2805  ; 083: BSC/R,SCR.CPU,FADCBR0,WAIT,NAD/028
;2806  ; 0E1: FADC/A,BSC/FAL.HL,OPLD/MC1,NAD/0E2
;2807  ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;2808  ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2809  ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2810  ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D4
;2811  ; 0D2: FADC/A,BSC/NL,BSC/FAL.LH,WAIT,NAD/0D3
;2812  ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2813  ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2814  ;
;2815  ;
;2816  ; LOGIC DESCRIPTION
;2817  ;
;2818  ; ALL OF THIS LOGIC IS ON THE FCT MODULE.
;2819  ;
;2820  ; ERROR DESCRIPTION
;2821  ; RECEIVED SIGN RESULT
;2822  ;
;2823  ; SYNC POINT DESCRIPTION
;2824  ;
;2825  ;--
;2826  ;*****************************************************************
;2827  ;/////////////////////////////////////////////
;2828  ;+
;2829  ; SUBTEST 1
;2830  ; (SB(1)H)*(A<B)*[-(POLYADDL+(-IR2H)]*(-A OUT OF NORM L)
;2831  ;-
;2832  =0
U 10EA, 0018.0039.0980.09F8.0000.11F8  ;2833 SGN6: NEWTST[.2]
```

```
U 10EB, 0000,003C,0180,0800,0000,10EC  ;2834  NOP
U 10EC, 0000,003D,0180,0800,0000,120F  ;2835 =0 SUBTEST
U 10ED, 0000,003C,0180,0800,0000,10EE  ;2836  NOP
U 10EE, 0000,003D,0180,0800,0000,120A  ;2837 =0 ERLOOP
U 10EF, 0000,003C,0180,0800,0000,12E7  ;2838  NOP
U 12E7, 0018,0038,4580,09F0,0000,10F0  ;2839  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 10F0, 0000,003D,0180,0800,0000,1244  ;2840 =0 CALL[LDDAT3]  ; LA<LB,SA=0,SB=1
U 10F1, 0000,003C,0180,0800,0000,10F2  ;2841  NOP
U 10F2, 0000,003D,0180,0800,0000,125C  ;2842 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 10F3, 0000,003C,0180,0800,0000,10F4  ;2843  NOP
U 10F4, 2018,0039,0180,0800,4200,1233  ;2844 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 10F5, 0000,003C,0180,0800,0000,10F6  ;2845  NOP
U 10F6, 0000,003D,0180,0800,0000,122F  ;2846 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 10F7, 0000,007C,0180,0800,0000,12E8  ;2847  CPSYNC
U 12E8, 0000,003C,01D8,0800,0000,12E9  ;2848  Q_ACC
U 12E9, 0019,2034,45C0,0800,0000,12EA  ;2849  Q_Q.AND.K[.8000]
U 12EA, 0001,203C,0180,09E8,0000,12EB  ;2850  RC[0D]_Q
U 12EB, 0011,2020,0180,0970,0010,12EC  ;2851  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12EC, 0000,013C,0180,0800,0000,10F8  ;2852  Z?
U 10F8, 0000,003D,0180,0800,0000,120B  ;2853 =0 ERROR1
U 10F9, 0000,003C,0180,0800,0000,10FA  ;2854  NOP
          ;2855 ;/////////////////////////////////////////
          ;2856 ; SUBTEST 2
          ;2857 ; (-SB(1)H)*(A<B)*[-(POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
          ;2858 ;-
          ;2859 =0
U 10FA, 0000,003D,0180,0800,0000,120F  ;2860  SUBTEST
U 10FB, 0000,003C,0180,0800,0000,10FC  ;2861  NOP
U 10FC, 0000,003D,0180,0800,0000,120A  ;2862 =0 ERLOOP
U 10FD, 0000,003C,0180,0800,0000,12ED  ;2863  NOP
U 12ED, 0003,003C,0180,09F0,0000,10FE  ;2864  RC[0E]_0  ; EXPECTED SIGN RESULT
U 10FE, 0000,003D,0180,0800,0000,1242  ;2865 =0 CALL[LDDAT2]  ; LA<LB,SA=0,SB=0
U 10FF, 0000,003C,0180,0800,0000,110A  ;2866  NOP
U 110A, 0000,003D,0180,0800,0000,125C  ;2867 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 110B, 0000,003C,0180,0800,0000,1110  ;2868  NOP
U 1110, 2018,0039,0180,0800,4200,1233  ;2869 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1111, 0000,003C,0180,0800,0000,1112  ;2870  NOP
U 1112, 0000,003D,0180,0800,0000,122F  ;2871 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1113, 0000,007C,0180,0800,0000,12EE  ;2872  CPSYNC
U 12EE, 0000,003C,01D8,0800,0000,12EF  ;2873  Q_ACC
U 12EF, 0019,2034,45C0,0800,0000,12F0  ;2874  Q_Q.AND.K[.8000]
U 12F0, 0001,203C,0180,09E8,0000,12F1  ;2875  RC[0D]_Q
U 12F1, 0011,2020,0180,0970,0010,12F2  ;2876  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12F2, 0000,013C,0180,0800,0000,1114  ;2877  Z?
U 1114, 0000,003D,0180,0800,0000,120B  ;2878 =0 ERROR1
U 1115, 0000,003C,0180,0800,0000,1116  ;2879  NOP
          ;2880 ;/////////////////////////////////////////
          ;2881 ;*
          ;2882 ; SUBTEST 3
          ;2883 ; (SB(1)H)*(A>B)*[-(POLYADDL)+(-IR2H)]*(-A OUT OF NORM L)
          ;2884 ;-
U 1116, 0000,003D,0180,0800,0000,120F  ;2885 =0 SUBTEST
U 1117, 0000,003C,0180,0800,0000,1118  ;2886  NOP
U 1118, 0000,003D,0180,0800,0000,120A  ;2887 =0 ERLOOP
U 1119, 0000,003C,0180,0800,0000,12F3  ;2888  NOP
```

```
U 12F3. 0003.003C.0180.09F0.0000.111A  ;2889  RC[0E]_0  ; EXPECTED SIGN RESULT
U 111A. 0000.003D.0180.0800.0000.1240  ;2890 =0 CALL[LDDATA]  ; LA>LB,SA=0,SB=1
U 111B. 0000.003C.0180.0800.0000.111C  ;2891  NOP
U 111C. 0000.003D.0180.0800.0000.125C  ;2892 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 111D. 0000.003C.0180.0800.0000.111E  ;2893  NOP
U 111E. 2018.0039.0180.0800.4200.1233  ;2894 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 111F. 0000.003C.0180.0800.0000.1130  ;2895  NOP
U 1130. 0000.003D.0180.0800.0000.122F  ;2896 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1131. 0000.007C.0180.0800.0000.12F4  ;2897  CPSYNC
U 12F4. 0000.003C.01D8.0800.0000.12F5  ;2898  Q_ACC
U 12F5. 0019.2034.45C0.0800.0000.12F6  ;2899  Q_Q.AND.K[.8000]
U 12F6. 0001.203C.0180.09E8.0000.12F7  ;2900  RC[0D]_Q
U 12F7. 0011.2020.0180.0970.0010.12F8  ;2901  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12F8. 0000.013C.0180.0800.0000.1132  ;2902  Z?
U 1132. 0000.003D.0180.0800.0000.120B  ;2903 =0 ERROR1
U 1133. 0000.003C.0180.0800.0000.1134  ;2904  NOP
          :2905 ;/////////////////////////////////////////
          :2906 ;*
          :2907 ; SUBTEST 4
          :2908 ; (SB(1)H)*(A<B)*[-(POLYADDL)+(IR2H)]*(-A OUT OF NORM L)
          :2909 ;-
U 1134. 0000.003D.0180.0800.0000.120F  ;2910 =0 SUBTEST
U 1135. 0000.003C.0180.0800.0000.1136  ;2911  NOP
U 1136. 0000.003D.0180.0800.0000.120A  ;2912 =0 ERLOOP
U 1137. 0000.003C.0180.0800.0000.12F9  ;2913  NOP
U 12F9. 0003.003C.0180.09F0.0000.1138  ;2914  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1138. 0000.003D.0180.0800.0000.124A  ;2915 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 1139. 0000.003C.0180.0800.0000.113A  ;2916  NOP
U 113A. 0000.003D.0180.0800.0000.125C  ;2917 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 113B. 0000.003C.0180.0800.0000.113C  ;2918  NOP
U 113C. 2018.0039.6580.0800.4200.1233  ;2919 =0 ALU_K[.10],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS DIVF R,R
U 113D. 0000.003C.0180.0800.0000.113E  ;2920  NOP
U 113E. 0000.003D.0180.0800.0000.122F  ;2921 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 113F. 0000.007C.0180.0800.0000.12FA  ;2922  CPSYNC
U 12FA. 0000.003C.01D8.0800.0000.12FB  ;2923  Q_ACC
U 12FB. 0019.2034.45C0.0800.0000.12FC  ;2924  Q_Q.AND.K[.8000]
U 12FC. 0001.203C.0180.09E8.0000.12FD  ;2925  RC[0D]_Q
U 12FD. 0011.2020.0180.0970.0010.12FE  ;2926  ALU_Q.XOR.RC[0E],CLK.UBCC
U 12FE. 0000.013C.0180.0800.0000.1140  ;2927  Z?
U 1140. 0000.003D.0180.0800.0000.120B  ;2928 =0 ERROR1
U 1141. 0000.003C.0180.0800.0000.1142  ;2929  NOP
          :2930
          :2931
          :2932 ;/////////////////////////////////////////
          :2933 ;*
          :2934 ; SUBTEST 5
          :2935 ; (-SB(1)H)*(A<B)*[-(POLYADDL+(-IR2H))]*(A OUT OF NORM L)
          :2936 ; ALU SUBTRACT AND EXPONENT DIFFERENCE IS EQUAL TO (1) LB=LA+1.
          :2937 ; EXPECTED SIGN RESULT LO.
          :2938 ;-
U 1142. 0000.003D.0180.0800.0000.120F  ;2939 =0 SUBTEST
U 1143. 0000.003C.0180.0800.0000.1144  ;2940  NOP
U 1144. 0000.003D.0180.0800.0000.120A  ;2941 =0 ERLOOP
U 1145. 0003.003C.0180.09F0.0000.1146  ;2942  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1146. 0000.003D.0180.0800.0000.12B4  ;2943 =0 CALL[AONORM]  ; ASSERT A OUT OF NORM
```

```
U 1147. 0000.003C.0180.0800.0000.1148  ;2944  NOP   ; LATCH AON ON THE FCT.
U 1148. 0000.003D.0180.0800.0000.1048  ;2945 =0 CALL[LD.BR]  ; BR1_0.1111111..
U 1149. 0000.003C.0180.0800.0000.114A  ;2946  NOP   ; BR0_0
U 114A. 0000.003D.0180.0800.0000.12A0  ;2947 =0 CALL[EXP.D2]  ; LA_34,SA_1
U 114B. 0000.003C.0180.0800.0000.114C  ;2948  NOP   ; LB_35,SB_0
U 114C. 0000.003D.0180.0800.0000.125C  ;2949 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 114D. 0000.003C.0180.0800.0000.114E  ;2950  NOP
U 114E. 2018.0039.0180.0800.4200.1233  ;2951 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 114F. 0000.003C.0180.0800.0000.1150  ;2952  NOP
U 1150. 0000.003D.0180.0800.0000.122F  ;2953 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1151. 0000.007C.0180.0800.0000.12FF  ;2954  CPSYNC
U 12FF. 0000.003C.01D8.0800.0000.1300  ;2955  Q_ACC
U 1300. 0019.2034.45C0.0800.0000.1301  ;2956  Q_Q.AND.K[.8000]
U 1301. 0001.203C.0180.09E8.0000.1302  ;2957  RC[0D]_Q
U 1302. 0011.2020.0180.0970.0010.1303  ;2958  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1303. 0000.013C.0180.0800.0000.1152  ;2959  Z?
U 1152. 0000.003D.0180.0800.0000.120B  ;2960 =0 ERROR1
U 1153. 0000.003C.0180.0800.0000.1156  ;2961  NOP
          ;2962
          ;2963 ;/////////////////////////////////////////////
          ;2964 ;*
          ;2965 ; SUBTEST 6
          ;2966 ; (SB(1)H)*(A=B)*[-(POLYADDL*(-IR2H))]*(A OUT OF NORM L)
          ;2967 ; EXPONENT DIFFERENCE EQUAL (0) LA=LB
          ;2968 ; EXPECTED SIGN RESULT HI.
          ;2969 ;-
U 1156. 0000.003D.0180.0800.0000.120F  ;2970 =0 SUBTEST
U 1157. 0000.003C.0180.0800.0000.1158  ;2971  NOP
U 1158. 0000.003D.0180.0800.0000.120A  ;2972 =0 ERLOOP
U 1159. 0018.0038.4580.09F0.0000.115A  ;2973  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 115A. 0000.003D.0180.0800.0000.12B4  ;2974 =0 CALL[AONORM]  ; ASSERT A OUT OF NORM
U 115B. 0000.003C.0180.0800.0000.115C  ;2975  NOP   ; LATCH AON ON THE FCT.
U 115C. 0000.003D.0180.0800.0000.1048  ;2976 =0 CALL[LD.BR]  ; BR1_0.1111111..
U 115D. 0000.003C.0180.0800.0000.115E  ;2977  NOP   ; BR0_0
U 115E. 0000.003D.0180.0800.0000.12AE  ;2978 =0 CALL[EXP.D4]  ; LA_34,SA_0
U 115F. 0000.003C.0180.0800.0000.1160  ;2979  NOP   ; LB_34,SB_1
U 1160. 0000.003D.0180.0800.0000.125C  ;2980 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 1161. 0000.003C.0180.0800.0000.1162  ;2981  NOP
U 1162. 2018.0039.0180.0800.4200.1233  ;2982 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 1163. 0000.003C.0180.0800.0000.1164  ;2983  NOP
U 1164. 0000.003D.0180.0800.0000.122F  ;2984 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1165. 0000.007C.0180.0800.0000.1304  ;2985  CPSYNC
U 1304. 0000.003C.01D8.0800.0000.1305  ;2986  Q_ACC
U 1305. 0019.2034.45C0.0800.0000.1306  ;2987  Q_Q.AND.K[.8000]
U 1306. 0001.203C.0180.09E8.0000.1307  ;2988  RC[0D]_Q
U 1307. 0011.2020.0180.0970.0010.1308  ;2989  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1308. 0000.013C.0180.0800.0000.1166  ;2990  Z?
U 1166. 0000.003D.0180.0800.0000.120B  ;2991 =0 ERROR1
U 1167. 0000.003C.0180.0800.0000.1168  ;2992  NOP
          ;2993 ;/////////////////////////////////////////////
          ;2994 ;*
          ;2995 ; SUBTEST 7
          ;2996 ; (-SB(1)H)*(A=B)*[-(POLYADDL*(-IR2H))]*(A OUT OF NORM L)
          ;2997 ; EXPONENT DIFFERENCE EQUAL (0) LA=LB
          ;2998 ; EXPECTED SIGN RESULT LO.
```

```
          ;2999 ;-
U 1168. 0000.003D.0180.0800.0000.120F  ;3000 =0 SUBTEST
U 1169. 0000.003C.0180.0800.0000.116A  ;3001  NOP
U 116A. 0000.003D.0180.0800.0000.120A  ;3002 =0 ERLOOP
U 116B. 0003.003C.0180.09F0.0000.116C  ;3003  RC[0E]_0  ; EXPECTED SIGN RESULT
U 116C. 0000.003D.0180.0800.0000.12B4  ;3004 =0 CALL[AONORM]  ; ASSERT A OUT OF NORM
U 116D. 0000.003C.0180.0800.0000.116E  ;3005  NOP  ; LATCH AON ON THE FCT.
U 116E. 0000.003D.0180.0800.0000.1048  ;3006 =0 CALL[LD.BR]  ; BR1_0.1111111..
U 116F. 0000.003C.0180.0800.0000.1170  ;3007  NOP  ; BR0_0
U 1170. 0000.003D.0180.0800.0000.1293  ;3008 =0 CALL[EXP.D0]  ; LA_34,SA_1
U 1171. 0000.003C.0180.0800.0000.1172  ;3009  NOP  ; LB_34,SB_0
U 1172. 0000.003D.0180.0800.0000.125C  ;3010 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 1173. 0000.003C.0180.0800.0000.1174  ;3011  NOP
U 1174. 2018.0039.0180.0800.4200.1233  ;3012 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADD R,R
U 1175. 0000.003C.0180.0800.0000.1176  ;3013  NOP
U 1176. 0000.003D.0180.0800.0000.122F  ;3014 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 1177. 0000.007C.0180.0800.0000.1309  ;3015  CPSYNC
U 1309. 0000.003C.01D8.0800.0000.130A  ;3016  Q_ACC
U 130A. 0019.2034.45C0.0800.0000.130B  ;3017  Q_Q.AND.K[.8000]
U 130B. 0001.203C.0180.09E8.0000.130C  ;3018  RC[0D]_Q
U 130C. 0011.2020.0180.0970.0010.130D  ;3019  ALU_Q.XOR.RC[0E],CLK.UBCC
U 130D. 0000.013C.0180.0800.0000.1178  ;3020  Z?
U 1178. 0000.003D.0180.0800.0000.120B  ;3021 =0 ERROR1
U 1179. 0000.003C.0180.0800.0000.117A  ;3022  NOP
          ;3023
```

```
                ;3024  .PAGE "TEST 283 SIGN LOGIC FOR SA.XOR.SX "
                ;3025 ;*************************************************************
                ;3026 ;**
                ;3027 ; TEST SIGN LOGIC FOR SA.XOR.SX
                ;3028 ;
                ;3029 ;
                ;3030 ; TEST DESCRIPTION
                ;3031 ;
                ;3032 ; THE SIGN DECISION LOGIC FOR SA.XOR.SX IS VERIFIED
                ;3033 ; IN THIS TEST. THE RESULTING SIGN OUPUT IS CHECKED
                ;3034 ; FOR EACH ONE OF THE FOUR INPUT COMBINATIONS.
                ;3035 ;
                ;3036 ; FPA UCODE USED
                ;3037 ; LOCATION CONTENTS
                ;3038 ; 0F4: BSC/NH,SGNC/A.XOR.X,NAD/PSTALL
                ;3039 ; 0F5: MSC/LDSX,NAD/PSTALL
                ;3040 ; 0F8: BSC/R,SCR/R.R.EAC/LDAB,SGNC/LDS,WAIT,NAD/0F6
                ;3041 ; 087: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,FADC/R1,WAIT,NAD
                ;3042 ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
                ;3043 ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/27
                ;3044 ;
                ;3045 ;
                ;3046 ; LOGIC DESCRIPTION
                ;3047 ;
                ;3048 ; ALL OF THIS LOGIC IS ON THE FCT.
                ;3049 ;
                ;3050 ; ERROR DESCRIPTION
                ;3051 ; EXPECTED SIGN RESULT
                ;3052 ; RECEIVED SIGN RESULT
                ;3053 ;
                ;3054 ; SYNC POINT DESCRIPTION
                ;3055 ;
                ;3056 ;--
                ;3057 ;*************************************************************
                ;3058 ;/////////////////////////////////////////////
                ;3059 ;*
                ;3060 ; SUBTEST 1
                ;3061 ; (SA(1)H).XOR.(SX(1)H)
                ;3062 ;-
                ;3063 =0
U 117A. 0018.0039.0980.09F8.0000.11F8  ;3064 SGN7: NEWTST[.2]
U 117B. 0000.003C.0180.0800.0000.117C  ;3065  NOP
U 117C. 0000.003D.0180.0800.0000.120F  ;3066 =0 SUBTEST
U 117D. 0018.0038.4580.0AA8.0000.117E  ;3067  R[S]_K[.8000]  ; SA=1
U 117E. 0000.003D.0180.0800.0000.125C  ;3068 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 117F. 0000.003C.0180.0800.0000.1180  ;3069  NOP
U 1180. 2018.0039.0180.0800.4200.1233  ;3070 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1181. 0000.003C.0180.0800.0000.1182  ;3071  NOP
U 1182. 0000.003D.0180.0800.0000.122F  ;3072 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 1183. 0003.003C.0180.09F0.0000.1184  ;3073  RC[0E]_0  ; EXPECTED SIGN RESULT
U 1184. 0000.003D.0180.0800.0000.122D  ;3074 =0 CALL[TRP.F5]  ; LOAD SX=1
U 1185. 0000.003C.0180.0800.0000.1186  ;3075  NOP
U 1186. 0000.003D.0180.0800.0000.122B  ;3076 =0 CALL[TRP.F4]  ; SA_SA.XOR.SX
U 1187. 0000.003C.01D8.0800.0000.130E  ;3077  Q_ACC
U 130E. 0019.2034.45C0.0800.0000.130F  ;3078  Q_Q.AND.K[.8000]
```

```
U 130F. 0001.203C.0180.09E8.0000.1310  ;3079  RC[0D]_Q
U 1310. 0011.2020.0180.0970.0010.1311  ;3080  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1311. 0000.013C.0180.0800.0000.1188  ;3081  Z?
U 1188. 0000.003D.0180.0800.0000.120B  ;3082 =0 ERROR1
U 1189. 0000.003C.0180.0800.0000.118A  ;3083  NOP
            ;3084 ;/////////////////////////////////////////
            ;3085 ;*
            ;3086 ; SUBTEST 2
            ;3087 ; (SA(1)H).XOR.(SX(1)H)
            ;3088 ;-
U 118A. 0000.003D.0180.0800.0000.120F  ;3089 =0 SUBTEST
U 118B. 0000.003C.0180.0800.0000.118C  ;3090  NOP
U 118C. 0000.003D.0180.0800.0000.120A  ;3091 =0 ERLOOP
U 118D. 0018.0038.4580.0AA8.0000.118E  ;3092  R[5]_K[.8000]  ; SA=1
U 118E. 0000.003D.0180.0800.0000.125C  ;3093 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 118F. 0000.003C.0180.0800.0000.1190  ;3094  NOP
U 1190. 2018.0039.0180.0800.4200.1233  ;3095 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 1191. 0000.003C.0180.0800.0000.1192  ;3096  NOP
U 1192. 0000.003D.0180.0800.0000.122F  ;3097 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 1193. 0018.0038.4580.09F0.0000.1194  ;3098  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 1194. 0000.003D.0180.0800.0000.122D  ;3099 =0 CALL[TRP.F5]  ; LOAD SX=1
U 1195. 0003.003C.0180.0AA8.0000.1196  ;3100  R[5]_0  ; FINAL SA=0
U 1196. 0000.003D.0180.0800.0000.1221  ;3101 =0 CALL[TRP.87]  ; LOAD SA,LA
U 1197. 0000.003C.0180.0A28.0000.1312  ;3102  LAB_R[5]  ; FPA AT 087
U 1312. 0000.007C.0180.0A28.0000.1198  ;3103  LAB_R[5],CPSYNC  ; GO TO 028
U 1198. 0000.003D.0180.0800.0000.122B  ;3104 =0 CALL[TRP.F4]  ; SA_SA.XOR.SX
U 1199. 0000.003C.01D8.0800.0000.1313  ;3105  Q_ACC
U 1313. 0019.2034.45C0.0800.0000.1314  ;3106  Q_Q.AND.K[.8000]
U 1314. 0001.203C.0180.09E8.0000.1315  ;3107  RC[0D]_Q
U 1315. 0011.2020.0180.0970.0010.1316  ;3108  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1316. 0000.013C.0180.0800.0000.119A  ;3109  Z?
U 119A. 0000.003D.0180.0800.0000.120B  ;3110 =0 ERROR1
U 119B. 0000.003C.0180.0800.0000.119C  ;3111  NOP
            ;3112 ;/////////////////////////////////////////
            ;3113 ;*
            ;3114 ; SUBTEST 3
            ;3115 ; (SA(1)H).XOR.(-SX(1)H)
            ;3116 ;-
U 119C. 0000.003D.0180.0800.0000.120F  ;3117 =0 SUBTEST
U 119D. 0000.003C.0180.0800.0000.119E  ;3118  NOP
U 119E. 0000.003D.0180.0800.0000.120A  ;3119 =0 ERLOOP
U 119F. 0003.003C.0180.0AA8.0000.11A0  ;3120  R[5]_0  ; SA=0
U 11A0. 0000.003D.0180.0800.0000.125C  ;3121 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 11A1. 0000.003C.0180.0800.0000.11A2  ;3122  NOP
U 11A2. 2018.0039.0180.0800.4200.1233  ;3123 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 11A3. 0000.003C.0180.0800.0000.11A4  ;3124  NOP
U 11A4. 0000.003D.0180.0800.0000.122F  ;3125 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 11A5. 0018.0038.4580.09F0.0000.11A6  ;3126  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 11A6. 0000.003D.0180.0800.0000.122D  ;3127 =0 CALL[TRP.F5]  ; LOAD SX=0
U 11A7. 0018.0038.4580.0AA8.0000.11A8  ;3128  R[5]_K[.8000]  ; FINAL SA=1
U 11A8. 0000.003D.0180.0800.0000.1221  ;3129 =0 CALL[TRP.87]  ; LOAD SA,LA
U 11A9. 0000.003C.0180.0A28.0000.1317  ;3130  LAB_R[5]  ; FPA AT 087
U 1317. 0000.007C.0180.0A28.0000.11AA  ;3131  LAB_R[5],CPSYNC  ; GO TO 028
U 11AA. 0000.003D.0180.0800.0000.122B  ;3132 =0 CALL[TRP.F4]  ; SA_SA.XOR.SX
U 11AB. 0000.003C.01D8.0800.0000.1318  ;3133  Q_ACC
```

```
U 1318. 0019.2034.45C0.0800.0000.1319  ;3134  Q_Q.AND.K[.8000]
U 1319. 0001.203C.0180.09E8.0000.131A  ;3135  RC[0D]_Q
U 131A. 0011.2020.0180.0970.0010.131B  ;3136  ALU_Q.XOR.RC[0E],CLK.UBCC
U 131B. 0000.013C.0180.0800.0000.11AC  ;3137  Z?
U 11AC. 0000.003D.0180.0800.0000.120B  ;3138 =0 ERROR1
U 11AD. 0000.003C.0180.0800.0000.11AE  ;3139  NOP
          ;3140 ;/////////////////////////////////////////
          ;3141 ;*
          ;3142 ; SUBTEST 4
          ;3143 ; (SA(1)H).XOR.(-SX(1)H)
          ;3144 ;-
U 11AE. 0000.003D.0180.0800.0000.120F  ;3145 =0 SUBTEST
U 11AF. 0000.003C.0180.0800.0000.11B0  ;3146  NOP
U 11B0. 0000.003D.0180.0800.0000.120A  ;3147 =0 ERLOOP
U 11B1. 0003.003C.0180.0AA8.0000.11B2  ;3148  R[5]_0    ; SA=0
U 11B2. 0000.003D.0180.0800.0000.125C  ;3149 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 11B3. 0000.003C.0180.0800.0000.11B4  ;3150  NOP
U 11B4. 2018.0039.0180.0800.4200.1233  ;3151 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 11B5. 0000.003C.0180.0800.0000.11B6  ;3152  NOP
U 11B6. 0000.003D.0180.0800.0000.122F  ;3153 =0 CALL[TRP.F8]  ; LOAD SA,SB,LA,LB
U 11B7. 0003.003C.0180.09F0.0000.11B8  ;3154  RC[0E]_0  ; EXPECTED SIGN RESULT
U 11B8. 0000.003D.0180.0800.0000.122D  ;3155 =0 CALL[TRP.F5]  ; LOAD SX=0
U 11B9. 0000.003C.0180.0800.0000.11BA  ;3156  NOP
U 11BA. 0000.003D.0180.0800.0000.122B  ;3157 =0 CALL[TRP.F4]  ; SA_SA.XOR.SX
U 11BB. 0000.003C.01D8.0800.0000.131C  ;3158  Q_ACC
U 131C. 0019.2034.45C0.0800.0000.131D  ;3159  Q_Q.AND.K[.8000]
U 131D. 0001.203C.0180.09E8.0000.131E  ;3160  RC[0D]_Q
U 131E. 0011.2020.0180.0970.0010.131F  ;3161  ALU_Q.XOR.RC[0E],CLK.UBCC
U 131F. 0000.013C.0180.0800.0000.11BC  ;3162  Z?
U 11BC. 0000.003D.0180.0800.0000.120B  ;3163 =0 ERROR1
U 11BD. 0000.003C.0180.0800.0000.11BE  ;3164  NOP
```

```
                ;3165  .PAGE "TEST 284 SIGN LOGIC FOR POLYADDL AND ADD/SUB"
                ;3166 ;*************************************************************
                ;3167 ;**
                ;3168 ; TEST SIGN LOGIC FOR POLYADDL AND ADD/SUB
                ;3169 ;
                ;3170 ; TEST DESCRIPTION
                ;3171 ;
                ;3172 ; THIS TEST CHECKS THE POLYADDL, IR2H INPUT LOGIC
                ;3173 ; COMBINATIONS NOT VERIFIED IN PREVIOUS TESTS.
                ;3174 ; POLYADDL NOT STUCK AT (0) WAS VERIFIED IN SIGN
                ;3175 ; PROCESSOR TEST 5, SUBTEST 6.
                ;3176 ;
                ;3177 ; FPA UCODE USED
                ;3178 ; LOCATION CONTENTS
                ;3179 ; 0F3: BSC/NH,SGNC/A.RES,MISC/P.ADD,NAD/PSTALL
                ;3180 ; 0F6: BSC/NH,SGNC/A.RES,NAD/PSTALL
                ;3181 ; 0F8: BSC/R,SCR/R.R,EAC/LDAB,SGNC/LDS,WAIT,NAD/0F6
                ;3182 ; 025: BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/26
                ;3183 ; 026: AMXC/LA,EALUC/A,EAC/LDPR,NAD/27
                ;3184 ;
                ;3185 ; LOGIC DESCRIPTION
                ;3186 ;
                ;3187 ; ALL OF THS LOGIC IS ON THE FCT MODULE.
                ;3188 ;
                ;3189 ; ERROR DESCRIPTION
                ;3190 ; EXPECTED SIGN RESULT
                ;3191 ; RECEIVED SIGN RESULT
                ;3192 ;
                ;3193 ; SYNC POINT DESCRIPTION
                ;3194 ;
                ;3195 ;--
                ;3196 ;*************************************************************************
                ;3197 ;/////////////////////////////////////////////
                ;3198 ;*
                ;3199 ; SUBTEST 1
                ;3200 ; (POLYADDL)*(IR2H)
                ;3201 ;-
                ;3202 =0
U 11BE, 0018,0039,0980,09F8,0000,11F8  ;3203 SGN8: NEWTST[.2]
U 11BF, 0000,003C,0180,0800,0000,11C0  ;3204  NOP
U 11C0, 0000,003D,0180,0800,0000,120F  ;3205 =0 SUBTEST
U 11C1, 0000,003C,0180,0800,0000,11C2  ;3206  NOP
U 11C2, 0000,003D,0180,0800,0000,120A  ;3207 =0 ERLOOP
U 11C3, 0018,0038,4580,09F0,0000,11C4  ;3208  RC[0E],K[.8000]  ; EXPECTED SIGN RESULT
U 11C4, 0000,003D,0180,0800,0000,124A  ;3209 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 11C5, 0000,003C,0180,0800,0000,11C6  ;3210  NOP
U 11C6, 0000,003D,0180,0800,0000,125C  ;3211 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 11C7, 0000,003C,0180,0800,0000,11C8  ;3212  NOP
U 11C8, 2018,0039,2180,0800,4200,1233  ;3213 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS MULF R,R
U 11C9, 0000,003C,0180,0800,0000,11CA  ;3214  NOP
U 11CA, 0000,003D,0180,0800,0000,122F  ;3215 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 11CB, 0000,003C,0180,0800,0000,11CC  ;3216  NOP
U 11CC, 0000,003D,0180,0800,0000,1229  ;3217 =0 CALL[TRP.F3]  ; ASSERT POLYADDL
U 11CD, 0000,003C,01D8,0800,0000,1320  ;3218  Q_ACC
U 1320, 0019,2034,45C0,0800,0000,1321  ;3219  Q_Q.AND.K[.8000]
```

```
U 1321. 0001.203C.0180.09E8.0000.1322  ;3220  RC[0D]_Q
U 1322. 0011.2020.0180.0970.0010.1323  ;3221  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1323. 0000.013C.0180.0800.0000.11CE  ;3222  Z?
U 11CE. 0000.003D.0180.0800.0000.120B  ;3223 =0 ERROR1
U 11CF. 0000.003C.0180.0800.0000.11D0  ;3224  NOP
           ;3225 ;/////////////////////////////////////////
           ;3226 ;*
           ;3227 ; SUBTEST 2
           ;3228 ; (-POLYADDL)*(-IR2H)
           ;3229 ;-
U 11D0. 0000.003D.0180.0800.0000.120F  ;3230 =0 SUBTEST
U 11D1. 0000.003C.0180.0800.0000.11D2  ;3231  NOP
U 11D2. 0000.003D.0180.0800.0000.120A  ;3232 =0 ERLOOP
U 11D3. 0018.0038.4580.09F0.0000.11D4  ;3233  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 11D4. 0000.003D.0180.0800.0000.124A  ;3234 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 11D5. 0000.003C.0180.0800.0000.11D6  ;3235  NOP
U 11D6. 0000.003D.0180.0800.0000.125C  ;3236 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 11D7. 0000.003C.0180.0800.0000.11D8  ;3237  NOP
U 11D8. 2018.0039.0180.0800.4200.1233  ;3238 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 11D9. 0000.003C.0180.0800.0000.11DA  ;3239  NOP
U 11DA. 0000.003D.0180.0800.0000.122F  ;3240 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 11DB. 0000.007C.0180.0800.0000.1324  ;3241  CPSYNC
U 1324. 0000.003C.01D8.0800.0000.1325  ;3242  Q_ACC
U 1325. 0019.2034.45C0.0800.0000.1326  ;3243  Q_Q.AND.K[.8000]
U 1326. 0001.203C.0180.09E8.0000.1327  ;3244  RC[0D]_Q
U 1327. 0011.2020.0180.0970.0010.1328  ;3245  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1328. 0000.013C.0180.0800.0000.11DC  ;3246  Z?
U 11DC. 0000.003D.0180.0800.0000.120B  ;3247 =0 ERROR1
U 11DD. 0000.003C.0180.0800.0000.11DE  ;3248  NOP
           ;3249 ;/////////////////////////////////////////
           ;3250 ;*
           ;3251 ; SUBTEST 3
           ;3252 ; (POLYADDL)*(-IR2H)
           ;3253 ;-
U 11DE. 0000.003D.0180.0800.0000.120F  ;3254 =0 SUBTEST
U 11DF. 0000.003C.0180.0800.0000.11E0  ;3255  NOP
U 11E0. 0000.003D.0180.0800.0000.120A  ;3256 =0 ERLOOP
U 11E1. 0018.0038.4580.09F0.0000.11E2  ;3257  RC[0E]_K[.8000]  ; EXPECTED SIGN RESULT
U 11E2. 0000.003D.0180.0800.0000.124A  ;3258 =0 CALL[LDDAT6]  ; LA<LB,SA=1,SB=1
U 11E3. 0000.003C.0180.0800.0000.11E4  ;3259  NOP
U 11E4. 0000.003D.0180.0800.0000.125C  ;3260 =0 CALL[LD.PR]  ; SET EALU <7:0> NON ZERO
U 11E5. 0000.003C.0180.0800.0000.11E6  ;3261  NOP
U 11E6. 2018.0039.0180.0800.4200.1233  ;3262 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS ADDF R,R
U 11E7. 0000.003C.0180.0800.0000.11E8  ;3263  NOP
U 11E8. 0000.003D.0180.0800.0000.122F  ;3264 =0 CALL[TRP.F8]  ; LOAD LA,LB,SA,SB
U 11E9. 0000.003C.0180.0800.0000.11EA  ;3265  NOP
U 11EA. 0000.003D.0180.0800.0000.1229  ;3266 =0 CALL[TRP.F3]  ; ASSERT POLYADDL
U 11EB. 0000.003C.01D8.0800.0000.1329  ;3267  Q_ACC
U 1329. 0019.2034.45C0.0800.0000.132A  ;3268  Q_Q.AND.K[.8000]
U 132A. 0001.203C.0180.09E8.0000.132B  ;3269  RC[0D]_Q
U 132B. 0011.2020.0180.0970.0010.132C  ;3270  ALU_Q.XOR.RC[0E],CLK.UBCC
U 132C. 0000.013C.0180.0800.0000.11EC  ;3271  Z?
U 11EC. 0000.003D.0180.0800.0000.120B  ;3272 =0 ERROR1
U 11ED. 0000.003C.0180.0800.0000.11EE  ;3273  NOP
           ;3274
```

```
            ;3275  .PAGE "TEST 285 RESERVED"
            ;3276 =0
U 11EE. 0000,003D,0180,0800,0000,11F8  ;3277 TST16G: NEWTST
U 11EF. 0000,003C,0180,0800,0000,11F0  ;3278  NOP
            ;3279
```

```
            ;3280  .PAGE "TEST 286 RESERVED"
            ;3281 =0
U 11F0, 0000,003D,0180,0800,0000,11F8  ;3282 T286: NEWTST
U 11F1, 0000,003C,0180,0800,0000,132D  ;3283  NOP
            ;3284
```

```
            ;3285  .PAGE "DUMMY NEWTST"
            ;3286
U 132D, 0000,003D,0180,0800,0000,11F8  ;3287 END16: NEWTST
            ;3288
            ;3289
            ;3290 ; THE FOLLOWING OPCODE DATA IS REQUIRED FOR THIS TEST.
            ;3291 ;% 8
            ;3292 ;$ 5540, 0056
            ;3293 ;$ 5542, 0056
            ;3294 ;$ 5546, 0056
            ;3295 ;$ 5544, 0056
            ;3296 ;$ 5A40, 005B
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2692
ESKAR62.MCR                    MICRO2  1P(00)    26-JUL-85  17:07:18     Page   8
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 1918= 1919=
U 1008  2016= 2017= 1875 2005= 2045= 2046= 1876 2006=
U 1010  2049= 2050= 2118= 2119= 2121= 2122= 2128= 2129=
U 1018  2131= 2132= 2138= 2139= 2141= 2142= 2149= 2150=
U 1020  2158= 2159= 2162= 2163= 2170= 2171= 2179= 2180=
U 1028  2189= 2190= 2191= 2192= 2196= 2197= 2219= 2220=
U 1030  2239= 2240= 2242= 2243= 2244= 2245= 2247= 2248=
U 1038  2262= 2263= 2298= 2299= 2323= 2324= 2325= 2326=
U 1040  2345= 2346= 2360= 2361= 2439= 2440= 2441= 2442=
U 1048  2450= 2451= 2452= 2453= 2542= 2543= 1939: 1877
U 1050  2544= 2545= 2546= 2547= 2548= 2549= 2550= 2551=
U 1058  2552= 2553= 1942: 1878 2554= 2555= 1945: 1879
U 1060  2556= 2557= 2563= 2564= 2570= 2571= 2572= 2573=
U 1068  2574= 2575= 2576= 2577= 2578= 2579= 2580= 2581=
U 1070  2582= 2583= 2584= 2585= 2591= 2592= 2598= 2599=
U 1078  2600= 2601= 2602= 2603= 2604= 2605= 2606= 2607=
U 1080  2608= 2609= 2610= 2611= 2612= 2613= 2619= 2620=
U 1088  2626= 2627= 2628= 2629= 2630= 2631= 2632= 2633=
U 1090  2634= 2635= 2636= 2637= 2638= 2639= 2640= 2641=
U 1098  2647= 2648= 2654= 2655= 2656= 2657= 2658= 2659=
U 10A0  2660= 2661= 2662= 2663= 2665= 2666= 2667= 2668=
U 10A8  2669= 2670= 2676= 2677= 2683= 2684= 2685= 2686=
U 10B0  2687= 2688= 2689= 2690= 2691= 2692= 2693= 2694=
U 10B8  2695= 2696= 2697= 2698= 2704= 2705= 2713= 2714=
U 10C0  2715= 2716= 2717= 2718= 2719= 2720= 2721= 2722=
U 10C8  2723= 2724= 2725= 2726= 2727= 2728= 2729= 2730=
U 10D0  2732= 2733= 2739= 2740= 2750= 2751= 2752= 2753=
U 10D8  2754= 2755= 2756= 2757= 2758= 2759= 2760= 2761=
U 10E0  2762= 2763= 2764= 2765= 2766= 2767= 2769= 2770=
U 10E8  2776= 2777= 2833= 2834= 2835= 2836= 2837= 2838=
U 10F0  2840= 2841= 2842= 2843= 2844= 2845= 2846= 2847=
U 10F8  2853= 2854= 2860= 2861= 2862= 2863= 2865= 2866=
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 1880 2867= 2868= 1870: 1871: 1872: 1873:
U 1110  2869= 2870= 2871= 2872= 2878= 2879= 2885= 2886=
U 1118  2887= 2888= 2890= 2891= 2892= 2893= 2894= 2895=
U 1120  2461: 2462: 2463: 2464: 2465: 2466: 2467: 2468:
U 1128  2469: 2470: 2471: 2472: 2473: 2474: 2475: 2476:
U 1130  2896= 2897= 2903= 2904= 2910= 2911= 2912= 2913=
U 1138  2915= 2916= 2917= 2918= 2919= 2920= 2921= 2922=
U 1140  2928= 2929= 2939= 2940= 2941= 2942= 2943= 2944=
U 1148  2945= 2946= 2947= 2948= 2949= 2950= 2951= 2952=
U 1150  2953= 2954= 2960= 2961= 1900 1965: 2970= 2971=
U 1158  2972= 2973= 2974= 2975= 2976= 2977= 2978= 2979=
U 1160  2980= 2981= 2982= 2983= 2984= 2985= 2991= 2992=
U 1168  3000= 3001= 3002= 3003= 3004= 3005= 3006= 3007=
U 1170  3008= 3009= 3010= 3011= 3012= 3013= 3014= 3015=
U 1178  3021= 3022= 3064= 3065= 3066= 3067= 3068= 3069=
U 1180  3070= 3071= 3072= 3073= 3074= 3075= 3076= 3077=
U 1188  3082= 3083= 3089= 3090= 3091= 3092= 3093= 3094=
U 1190  3095= 3096= 3097= 3098= 3099= 3100= 3101= 3102=
U 1198  3104= 3105= 3110= 3111= 3117= 3118= 3119= 3120=
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  3121= 3122= 3123= 3124= 3125= 3126= 3127= 3128=
U 11A8  3129= 3130= 3132= 3133= 3138= 3139= 3145= 3146=
U 11B0  3147= 3148= 3149= 3150= 3151= 3152= 3153= 3154=
U 11B8  3155= 3156= 3157= 3158= 3163= 3164= 3203= 3204=
U 11C0  3205= 3206= 3207= 3208= 3209= 3210= 3211= 3212=
U 11C8  3213= 3214= 3215= 3216= 3217= 3218= 3223= 3224=
U 11D0  3230= 3231= 3232= 3233= 3234= 3235= 3236= 3237=
U 11D8  3238= 3239= 3240= 3241= 3247= 3248= 3254= 3255=
U 11E0  3256= 3257= 3258= 3259= 3260= 3261= 3262= 3263=
U 11E8  3264= 3265= 3266= 3267= 3272= 3273= 3277= 3278=
U 11F0  3282= 3283= 1901 1902 1903 1904 1905 1906
U 11F8  1916 1917 1920 1921 1922 1923 1924 1925
U 1200  1926 1927 1928 1929 1930 1931 1932 1933
U 1208  1934 1935 1936 1937 1938 1940 1941 1973
U 1210  1974 1980 1981 1982 2001 2002 2003 2004
U 1218  2014 2015 2018 2025 2026 2028 2029 2033
U 1220  2034 2036 2037 2038 2040 2041 2042 2052
U 1228  2053 2059 2060 2063 2064 2067 2068 2072
U 1230  2073 2075 2076 2086 2087 2088 2089 2090
U 1238  2091 2092 2104 2105 2106 2107 2111 2112
U 1240  2117 2120 2127 2130 2137 2140 2143 2148
U 1248  2151 2152 2156 2157 2160 2161 2164 2169
U 1250  2172 2173 2178 2181 2193 2194 2195 2198
U 1258  2199 2205 2206 2207 2217 2218 2221 2222
U 1260  2223 2224 2237 2238 2241 2246 2249 2264
U 1268  2265 2266 2273 2274 2275 2276 2277 2278
U 1270  2279 2280 2281 2282 2300 2301 2302 2303
U 1278  2304 2305 2306 2307 2308 2309 2310 2311
U 1280  2327 2328 2329 2330 2331 2332 2351 2352
U 1288  2353 2354 2356 2357 2362 2363 2364 2365
U 1290  2366 2367 2368 2379 2380 2381 2382 2383
U 1298  2384 2389 2390 2391 2392 2393 2394 2395
U 12A0  2399 2400 2401 2402 2403 2404 2408 2409
U 12A8  2410 2411 1966: 2412 2413 2414 2418 2419
U 12B0  2420 2421 2422 2423 2435 2436 2437 2438
U 12B8  2454 2455 2456 2457 2558 2559 2560 2561
U 12C0  2562 2586 2587 2588 2589 2590 2614 2615
U 12C8  2616 2617 2618 2642 2643 2644 2645 2646
U 12D0  2664 2671 2672 2673 2674 2675 2699 2700
U 12D8  2701 2702 2703 2731 2734 2735 2736 2737
U 12E0  2738 2768 2771 2772 2773 2774 2775 2839
U 12E8  2848 2849 2850 2851 2852 2864 2873 2874
U 12F0  2875 2876 2877 2889 2898 2899 2900 2901
U 12F8  2902 2914 2923 2924 2925 2926 2927 2955
U 1300  2956 2957 2958 2959 2986 2987 2988 2989
U 1308  2990 3016 3017 3018 3019 3020 3078 3079
U 1310  3080 3081 3103 3106 3107 3108 3109 3131
U 1318  3134 3135 3136 3137 3159 3160 3161 3162
U 1320  3219 3220 3221 3222 3242 3243 3244 3245
U 1328  3246 3268 3269 3270 3271 3287
U 1330 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 2694
ESKAR62.MCR                    MICRO2  1P(00)    26-JUL-85  17:07:18          Page   8

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR62        830

Used              830
Remaining

Total microwords used in memory U: 830
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2695
ESKAR62.MCR                    MICRO2  1P(00)    26-JUL-85  17:07:18     Page   8


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR62.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
ESKAR63.MCR MICRO2 1P(00) 26-JUL-85 17 SEQ 2696

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2136 TEST 287 NORMALIZER SHIFT WITHIN RANGE TEST
; 2283 TEST 288 NORMALIZER ROM TEST (1)
; 2489 TEST 289 NORMALIZER ROM TEST (2)
; 2734 TEST 28A TEST EXPONENT OVERFLOW DUE TO ROUNDING
; 3162 TEST 28B ACCELERATOR OVERRIDE TEST (1)
; 3245 TEST 28C ACCELERATOR OVERRIDE TEST (2)
; 3330 TEST 28D ACCELERATOR OVERRIDE TEST (3)
; 3415 TEST 28E ACCELERATOR OVERRIDE TEST (4)
; 3501 TEST 28F RESERVED
; 3506 TEST 290 RESERVED
; 3511 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR63
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844  ;*
            ;1845  ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846  ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847  ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848  ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849  ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850  ;
            ;1851  ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852  ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853  ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854  ; R[0F] AND DO A RETURN0.
            ;1855  ;
            ;1856  ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857  ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858  ; TO THE CONSOLE.
            ;1859  ;
            ;1860  ;-
U 1100, 0000.003C.1980.0800.0084.7003   ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7003   ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7003   ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7003   ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7003   ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7003   ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7003   ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7003   ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7003   ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7003   ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D, 0000.003C.8980.0800.0084.7003   ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7003   ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003   ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.104F   ;1874 TRPH0: Q_ID[USTACK]
U 104F, 0C00.003C.01E0.0800.0000.105B   ;1875  Q_D,D_Q
U 105B, 0000.003C.8180.3C00.0000.105F   ;1876  ID[USTACK]_D
U 105F, 0C00.003C.01E0.0800.0000.10BE   ;1877  Q_D,D_Q
U 10BE, 0000.003C.01C0.0A78.0000.10BF   ;1878  Q_R[0F]
U 10BF, 0001.2024.0180.0800.0010.10C0   ;1879  ALU_Q.ANDNOT.MASK.CLK.UBCC ; WAS TRAP EXPECTED?
U 10C0, 0000.013C.0180.0800.0000.1002   ;1880  Z?     ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000   ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882  ;*
            ;1883  ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884  ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885  ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886  ;
            ;1887  ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888  ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889  ;       FAULT STATUS REGISTER
            ;1890  ;       SBI ERROR REGISTER
            ;1891  ;       TIMEOUT ADDRESS REGISTER
            ;1892  ;       MAINTENANCE REGISTER
            ;1893  ;       CACHE PARITY REGISTER
            ;1894  ;       TB ERROR REGISTER 1
            ;1895  ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1014  ;1897 TRPH1: RC[0D]_SC
U 1014. 0000.003D.D980.0800.0084.70E1  ;1898 =0 SETRCF[.9]
U 1015. 0000.003C.49F0.2C00.0000.10C1  ;1899  Q_ID[TBER0]
U 10C1. 0001.203C.4DF0.2DB0.0000.10C2  ;1900  RC[6]_Q,Q_ID[TBER1]
U 10C2. 0001.203C.65F0.2DB8.0000.10C3  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 10C3. 0001.203C.6DF0.2DD8.0000.10C4  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 10C4. 0001.203C.75F0.2DE0.0000.10C5  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 10C5. 0001.203C.79F0.2DC8.0000.10C6  ;1904  RC[9]_Q,Q_ID[PARITY]
U 10C6. 0001.203C.69F0.2DC0.0000.10C7  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 10C7. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.10DB  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 10C8. 0000.003C.8580.0800.0084.70C9  ;1916 SCOPE: SC_K[.C]
U 10C9. 0803.001C.0180.0800.0000.1016  ;1917  ALU_NOT.MASK,D_ALU
U 1016. 0000.003D.7580.3C00.0000.1114  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1017. 0000.003C.65F8.0800.0084.70CA  ;1919  SC_K[.10],Q_0   ; SETUP TO WRITE IPL OF 1F
U 10CA. 0818.0038.8D80.0800.0000.10CB  ;1920  D_K[.1F]   ; ...
U 10CB. 0D00.003C.0180.0800.0000.10CC  ;1921  D_DAL.SC
U 10CC. 0000.003C.3D80.3C00.0000.10CD  ;1922  ID[PSL]_D   ; WRITE THE PSL
U 10CD. 0818.0038.0580.0800.0000.10CE  ;1923  D_K[.1]   ; SETUP TO CLEAR THE CES REGISTER
U 10CE. 0D00.003C.0180.0800.0000.10CF  ;1924  D_DAL.SC
U 10CF. 0F00.003C.3180.3C00.0000.10D0  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10D0. 0000.003C.0180.0800.0000.10D1  ;1926  NOP
U 10D1. 0000.003C.3980.3C00.0000.10D2  ;1927  ID[SIR]_D   ; CLEAR THE SIR REGISTER
U 10D2. 0000.003C.2980.3C00.0000.10D3  ;1928  ID[CLK.CS]_D   ; STOP INTERVAL TIMER
U 10D3. 0000.003C.4980.3C00.0000.10D4  ;1929  ID[TBER0]_D   ; SHUT OFF THE TB
U 10D4. 0818.0038.C180.0800.0000.10D5  ;1930  D_K[.FFFF]
U 10D5. 0000.003C.6580.3C00.0000.10D6  ;1931  ID[SBI.ERR]_D
U 10D6. 0F00.003C.6D80.3C00.0000.10D7  ;1932  ID[FAULT]_D,D_0
U 10D7. 0000.003C.6D80.3C00.0000.10D8  ;1933  ID[FAULT]_D
U 10D8. 0000.003C.6580.3C00.0000.10D9  ;1934  ID[SBI.ERR]_D
U 10D9. 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 10DA. 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 10DB. 0810.0038.0180.0978.0000.10DC  ;1937 ERROR1: D_RC[0F]
U 10DC. 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 10DD. 0810.0038.0180.0978.0000.10DE  ;1940 ERROR2: D_RC[0F]
U 10DE. 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2. 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155. 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA. 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 10DF. 0810.0038.4D80.0978.0000.10E0  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 10E0. 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 10E1. 0010.0038.01C0.0978.0000.10E2  ;1980 SETRCF: Q_RC[0F]
U 10E2. 0019.2034.4DC0.0800.0000.10E3  ;1981  Q_Q.AND.K[.FF00]
U 10E3. 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983
            ;1984 ;*
            ;1985 ; THIS SUBROUTINE ACCEPTS DATA IN REGISTERS 6,7,8,9,THEN TRAPS TO
            ;1986 ; FPA UCODE THAT LOADS THE FAD AR HI,AR LO,BR HI,BR LO.
            ;1987 ;-
            ;1988
            ;1989
U 10E4. 0818.0038.A580.0800.0000.10E5  ;1990 LDFAD: D_K[.60]
U 10E5. 0000.003C.0180.0800.0000.1018  ;1991  NOP
U 1018. 0819.0031.0180.0800.0000.1119  ;1992 =0 D_D.OR.K[.8],CALL[GENTRA]
U 1019. 0000.003C.5980.3C00.0000.10E6  ;1993  ID[ACC.2]_D
U 10E6. 0000.00FC.0380.0800.0000.10E7  ;1994  TRAP,FPA
U 10E7. 0000.003C.0180.0A28.0000.10E8  ;1995  LAB_R[5]    ; FPA AT 068.
U 10E8. 0000.007C.0180.0A28.0000.10E9  ;1996  LAB_R[5],CPSYNC    ; LOAD LA,SA
U 10E9. 0000.003C.0180.0A18.0000.10EA  ;1997  LAB_R[3]    ; FPA AT 069.
U 10EA. 0000.007C.0180.0A18.0000.10EB  ;1998  LAB_R[3],CPSYNC    ; LOAD LB,SB
U 10EB. 0000.003C.0180.0A30.0000.10EC  ;1999  LAB_R[6]    ; FPA AT 06A.
U 10EC. 0000.007C.0180.0A30.0000.10ED  ;2000  LAB_R[6],CPSYNC    ; AR1_R6
U 10ED. 0000.003C.0180.0A38.0000.10EE  ;2001  LAB_R[7]    ; FPA AT 06C.
U 10EE. 0000.007C.0180.0A38.0000.10EF  ;2002  LAB_R[7],CPSYNC    ; AR0_R7
U 10EF. 0000.003C.0180.0A40.0000.10F0  ;2003  LAB_R[8]    ; FPA AT 06D.
U 10F0. 0000.007C.0180.0A40.0000.10F1  ;2004  LAB_R[8],CPSYNC    ; BR1_R8
U 10F1. 0000.003C.0180.0A48.0000.10F2  ;2005  LAB_R[9]    ; FPA AT 083.
```

```
U 10F2, 0000.007C.0180.0A48.0000.10F3  ;2006  LAB_R[9],CPSYNC   ; BR0_R9
U 10F3, 0000.003E.0180.0800.0000.0001  ;2007  RETURN1    ; FPA AT 028. LOAD COMPLETE.
            ;2008
            ;2009 ;*
            ;2010 ; THIS SUBROUTINE TRAPS TO FPA UCODE THA EXECUTES A CONDITIONAL ADD
            ;2011 ; AND NORMALIZE USING THE OPERANDS LOADED BY THE LDFAD. THE FPA U CODE
            ;2012 ; TAKES THE FAD OUTPUT, LOADS THE NORMALIZER REGISTER (NR)  AND PERFORMS
            ;2013 ; THE NORMALIZE AND ROUND. THE CPU WAITS UNTILL THE FPA UCODE SIGNALS THAT
            ;2014 ; THE NORMALIZATION AND ROUND IS COMPLETE. THE RESULTS ARE THEN TRANSFERED
            ;2015 ; TO THE CPU FOR VERIFICATION.
            ;2016 ;-
            ;2017
            ;2018
            ;2019
            ;2020 =0
U 101A, 0818.0039.4180.0800.0000.1119  ;2021 NORM: D_K[.80],CALL[GENTRA]
U 101B, 0000.003C.5980.3C00.0000.10F4  ;2022  ID[ACC.2)_D
U 10F4, 0000.00FC.0380.0800.0000.10F5  ;2023  TRAP.FPA
U 10F5, 0000.003C.01D8.0800.0000.10F6  ;2024  Q_ACC    ; FPA AT 080.
            ;2025       ; DO FAD CONDITIONAL ADD/SUB
            ;2026       ; TEST EXPONENTS. BEN/5
            ;2027       ; NORMALIZE
U 10F6, 0000.003C.0180.0800.0000.10F7  ;2028  NOP    ; FPA AT 050.
            ;2029       ; UPDATE EXPONENT.
U 10F7, 0000.003C.0180.0800.0000.10F8  ;2030  NOP    ; FPA AT 051.
            ;2031       ; EXPONENT DIFF.>8
U 10F8, 0000.003C.0180.0800.0000.10F9  ;2032  NOP    ; FPA AT 058.
            ;2033       ; NR<63:52> CONTAIN A (1) ?
U 10F9, 0000.003C.0180.0800.0000.10FA  ;2034  NOP    ; FPA AT 059.
            ;2035       ; NR<63:52> CONTAIN A (1) ?
U 10FA, 0000.003C.0180.0800.0000.10FB  ;2036  NOP    ; FPA AT 05A.
            ;2037       ; NR<63:52> CONTAIN A (1) ?
U 10FB, 0000.003C.0180.0800.0000.10FC  ;2038  NOP    ; FPA AT 05B.
U 10FC, 0000.003C.0180.0800.0000.10FD  ;2039  NOP    ; FPA AT 05C.
            ;2040       ; COULD NOT NORMALIZE IN (60)
            ;2041       ; LEFT SHIFTS OF NR.
            ;2042       ; FORCE ZEROS ON THE BUS.
U 10FD, 0000.003C.0180.0800.0000.10FE  ;2043  NOP    ; FPA AT 05F.
            ;2044       ; NORMALIZATION OK.
            ;2045       ; UP DATE SIGN AND EXPONENT.
U 10FE, 0000.003C.0180.0800.0000.10FF  ;2046  NOP    ; FPA AT 056. ISSUE FPSYNC.
            ;2047       ; WAIT FOR CPSYNC.
            ;2048       ; OUTPUT RESULT HI
U 10FF, 0000.003C.0180.0800.0000.1109  ;2049  NOP    ; FPA AT 057.
            ;2050       ; OUTPUT RESULT LO
U 1109, 0000.003C.0180.0800.0000.110A  ;2051  NOP    ; WAIT (2) STATES FOR RESULTS
U 110A, 0000.003C.0180.0800.0000.110B  ;2052  NOP    ; TO STABILIZE.
U 110B, 0000.003C.01D8.0800.0000.1110  ;2053  Q_ACC    ; OUT PUT FRACTION HI, SIGN AND EXP.
U 1110, 0001.207C.0180.09E8.0000.1111  ;2054  RC[0D)_Q,CPSYNC   ; SAVE RESULT HI
U 1111, 0000.003C.01D8.0800.0000.1112  ;2055  Q_ACC    ; OUTPUT FRACTION LO
U 1112, 0001.203C.0180.09D8.0000.1113  ;2056  RC[0B)_Q    ; SAVE RESULT LO
U 1113, 0000.003E.0180.0800.0000.0001  ;2057  RETURN1    ; FPA AT 028.
            ;2058
            ;2059 ;
            ;2060 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
```

```
            ;2061 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;2062 ;   28: NAD/28  ; NOP
            ;2063 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;2064 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;2065 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;2066 ;  INITIALIZE ITSELF.
            ;2067 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;2068 ;
U 1114. 0818.0038.4580.0800.0000.1115  ;2069 FPINIT: D_K[.8000]
U 1115. 0000.003C.5D80.3C00.0000.101C  ;2070  ID[ACC.CS]_D  ; TURN ON FPA
U 101C. 0818.0039.2D80.0800.0000.1119  ;2071 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 101D. 0000.003C.5980.3C00.0000.1116  ;2072  ID[ACC.2]_D
U 1116. 0000.00FC.0380.0800.0000.101E  ;2073  TRAP.FPA  ; TRAP FPA TO 28.
U 101E. 0018.0039.1980.0800.0200.111F  ;2074 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 101F. 0F00.003C.0180.0800.0000.1020  ;2075  D_0
U 1020. 0000.003D.0180.0800.0000.1119  ;2076 =0 CALL,J/GENTRA
U 1021. 0000.003C.5980.3C00.0000.1117  ;2077  ID[ACC.2]_D
U 1117. 0000.00FC.0380.0800.0000.1022  ;2078  TRAP.FPA  ; TRAP FPA TO 0
U 1022. 0818.0039.2D80.0800.0000.1119  ;2079 =0 D_K[.28],CALL,J/GENTRA
U 1023. 0000.003C.5980.3C00.0000.1118  ;2080  ID[ACC.2]_D
U 1118. 0000.00FE.0380.0800.0000.0001  ;2081  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
            ;2082 ;
            ;2083 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2084 ;%0
            ;2085 ;$ 0000.0000. 0000.0000
            ;2086 ;
            ;2087 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2088 ; CONVERT IT INTO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2089 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2090 ;
U 1119. 0000.003C.65F8.0800.0084.711A  ;2091 GENTRA: SC_K[.10],Q_0
U 111A. 0D00.003C.8D80.0800.0084.711B  ;2092  D_DAL.SC,SC_K[.1F]
U 111B. 0801.001E.0180.0800.0000.0001  ;2093  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2094 ;
            ;2095 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2096 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2097 ;
            ;2098 =0
U 1024. 0F00.003D.0180.0800.0000.1119  ;2099 FPIRD: D_0,CALL,J/GENTRA
U 1025. 0000.003C.5980.3C00.0000.111C  ;2100  ID[ACC.2]_D
U 111C. 0000.00FC.0380.0800.0000.111D  ;2101  TRAP.FPA  ; TRAP TO FPA 0
U 111D. 0000.003C.0180.0800.0000.111E  ;2102  NOP
U 111E. 0000.003E.0180.0800.0000.0001  ;2103  RETURN1
            ;2104 ;
            ;2105 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2106 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2107 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2108 ;
U 111F. 2000.003C.0180.0800.0000.1120  ;2109 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 1120. 3000.003C.0180.6000.0000.1121  ;2110  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 1121. 0000.003C.0180.F800.0000.1122  ;2111 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1122. 0000.003C.0180.F800.0000.1123  ;2112  MCT/ALLOW.IB.READ
U 1123. 0000.003C.0180.F800.0000.1124  ;2113  MCT/ALLOW.IB.READ
U 1124. 0000.003C.0180.F800.0000.1125  ;2114  MCT/ALLOW.IB.READ
U 1125. 0000.003C.0180.F800.0000.1126  ;2115  MCT/ALLOW.IB.READ
```

```
U 1126, 1000,003C,0180,F800,0000,1127  ;2116  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1127, 0000,003E,0180,0800,0000,0001  ;2117  RETURN1
U 1128, 3000,003C,0180,0800,0000,1121  ;2118 WAITIB: IBC/START,J/IBW
          ;2119
          ;2120
          ;2121 =0
U 1026, 0818,0038,CD80,0800,0000,1027  ;2122 FADCHK: D_K[.F0]
U 1027, 0819,0030,F580,0800,0000,1028  ;2123  D_D.OR.K[.A]
U 1028, 0819,0015,0580,0800,0000,1119  ;2124 =0 D_D+K[.1],CALL[GENTRA]
U 1029, 0000,003C,5980,3C00,0000,1129  ;2125  ID[ACC.2]_D
U 1129, 0000,00FC,0380,0800,0000,112A  ;2126  TRAP.FPA
U 112A, 0000,003C,01D8,0800,0000,112B  ;2127  Q_ACC    ; FPA AT 0FB. BUSA_AR1
U 112B, 0001,207C,0180,09F0,0000,112C  ;2128  RC[0E]_Q,CPSYNC
U 112C, 0A00,003C,0180,0800,0000,112D  ;2129  D_ACC    ; FPA AT 0FC. BUSA_AR0
U 112D, 0001,007C,0180,09E8,0000,112E  ;2130  RC[0D]_D,CPSYNC
U 112E, 0000,003C,01D8,0800,0000,112F  ;2131  Q_ACC    ; FPA AT 0FD. BUSA_BR1
U 112F, 0001,207C,0180,09E0,0000,1130  ;2132  RC[0C]_Q,CPSYNC
U 1130, 0A00,003C,0180,0800,0000,1131  ;2133  D_ACC    ; FPA AT 0FE. BUSA_BR0
U 1131, 0001,003C,0180,09D8,0000,1132  ;2134  RC[0B]_D
U 1132, 0000,003E,0180,0800,0000,0001  ;2135  RETURN1
```

```
;2136   .PAGE "TEST 287 NORMALIZER SHIFT WITHIN RANGE TEST"
;2137  ;*****************************************************************
;2138  ;**
;2139  ; NORMALIZER SHIFT WITHIN RANGE TEST
;2140  ;
;2141  ; TEST DESCRIPTION
;2142  ;
;2143  ; THIS TEST USES OPERANDS IN AN ADDD R.R THAT WILL CAUSE
;2144  ; THE NORMALIZER TO DO (5) CONSECUTIVE SWR (SHIFT WITHIN RANGE)
;2145  ; DETERMINATIONS.
;2146  ;
;2147  ; FPA UCODE USED
;2148  ;
;2149  ; LOCATION CONTENTS
;2150  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2151  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2152  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2153  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2154  ; 06D: BSC/R,SCR/CPU/FADC/BR,WAIT,NAD/083
;2155  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2156  ; 028: NAD/028
;2157  ; 080: FADC/A.B,BSC/FAL.X,NRC/LD,AMXC/COND,EALUC/A,EAC/LDPR,
;2158  ;  BEN/5,NAD/050
;2159  ; 050: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC,P.ADD,EALUC/A+B,AMXC/PR,
;2160  ;  BMXC/NK,NAD/056
;2161  ; 051: FADC/A.B,BSC/FAL.HL,NRC/LD,NAD/050
;2162  ; 052: NRC/SL,SWR?,NAD/058
;2163  ; 058: NRC/SL,SWR?,NAD/059
;2164  ; 059: NRC/SL,SWR?,NAD/05A
;2165  ; 05A: NRC/SL,SWR?,NAD/05B
;2166  ; 05B: NRC/SL,SWR?,NAD/05C
;2167  ; 05C: BSC/NH,SGNC/A.RES,EALUC/K3FF,WAIT,FPSYNC,NAD/054
;2168  ; 05F: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC/P.ADD,EALUC/A+B,AMXC/PR,
;2169  ;  BMXC/NK,NAD/056
;2170  ; 056: BMXC/NK,BSC/NH,AMXC/PR,EALUC/A,WAIT,FPSYNC,NAD/057
;2171  ; 057: BMXC/NK,BSC/NL,AMXC/PR,EALUC/A,NAD/028
;2172  ; 054: BSC/NH,SGNC/A.RES,EALUC/K3FF,NAD/028
;2173  ;
;2174  ;
;2175  ; LOGIC DESCRIPTION
;2176  ;
;2177  ; THE NORMALIZER LOGIC IS ON THE FNM MODULE. THE NORMALIZER
;2178  ; INPUT TO UPDATE THE FINAL EXPONENT IS ON THE FCT MODULE.
;2179  ;
;2180  ;
;2181  ; ERROR DESCRIPTION
;2182  ;
;2183  ; EXPECTED NEXT ADDRESS
;2184  ; RECEIVED NEXT ADDRESS
;2185  ; SP1 HI
;2186  ; SP1 LO
;2187  ; SP2 HI
;2188  ; SP2 LO
;2189  ;
;2190  ; SYNC POINT DESCRIPTION
```

```
                ;2191 ;
                ;2192 ;--
                ;2193 ;*****************************************************************
                ;2194
                ;2195
                ;2196 =0
U 102A. 0018.0039.D580.09F8.0000.10C8  ;2197 SWR: NEWTST[.6]
U 102B. 0818.0038.3580.0800.0000.1133  ;2198  D_K[.50]
U 1133. 0019.0030.6180.09F0.0000.1134  ;2199  ALU_D.OR.K[.F],RC[0E]_ALU  ; SAVE EXPECTED NAD (05F)
U 1134. 0818.0038.4D80.0800.0000.1135  ;2200  D_K[.FF00]
U 1135. 0819.0030.4180.0800.0000.1136  ;2201  D_D.OR.K[.80]
U 1136. 0001.0028.0180.0AA0.0000.1137  ;2202  R[4]_NOT.D
U 1137. 0018.0038.6580.0800.0200.1138  ;2203  VA_K[.10]
U 1138. 0000.003C.0180.4000.0000.1139  ;2204  D[LONG]_CACHE
U 1139. 0001.003C.0180.0AA8.0000.113A  ;2205  R[5]_D      ; SAVE TO LOAD SA, LA.
U 113A. 0000.003C.01C0.0A20.0000.113B  ;2206  Q_R[4]
U 113B. 0001.003C.0180.09E0.0000.113C  ;2207  RC[0C]_D     ; SAVE
U 113C. 081D.0034.0180.0800.0000.113D  ;2208  ALU_D.AND.Q,D_ALU    ; MASK OUT SIGN AND EXPONENT
U 113D. 0001.003C.0180.0AB0.0000.113E  ;2209  R[6]_D      ; SAVE SP1 HI
U 113E. 0000.003C.0180.0803.0000.113F  ;2210  VA_VA+4
U 113F. 0000.003C.0180.4000.0000.1140  ;2211  D[LONG]_CACHE
U 1140. 0001.003C.0180.0AB8.0000.1141  ;2212  R[7]_D      ; SP1 LO
U 1141. 0001.003C.0180.09D8.0000.1142  ;2213  RC[0B]_D     ; SAVE
U 1142. 0000.003C.0180.0803.0000.1143  ;2214  VA_VA+4
U 1143. 0000.003C.0180.4000.0000.1144  ;2215  D[LONG]_CACHE
U 1144. 0001.003C.0180.0A98.0000.1145  ;2216  R[3]_D      ; SAVE TO LOAD SB, LB
U 1145. 0001.003C.0180.0AC0.0000.1146  ;2217  R[8]_D      ; SP2 HI
U 1146. 0001.003C.0180.09D0.0000.1147  ;2218  RC[0A]_D     ; SAVE
U 1147. 0000.003C.0180.0803.0000.1148  ;2219  VA_VA+4
U 1148. 0000.003C.0180.4000.0000.1149  ;2220  D[LONG]_CACHE
U 1149. 0001.003C.0180.0AC8.0000.114A  ;2221  R[9]_D      ; SP2 LO
U 114A. 0001.003C.0180.09C8.0000.114B  ;2222  RC[9]_D      ; SAVE
U 114B. 0018.0038.7580.0800.0200.102C  ;2223  VA_K[.20]
U 102C. 0000.003D.0180.0800.0000.111F  ;2224 =0 CALL[LOADIB]     ; OPCODE ADDD R,R
U 102D. 0000.003C.0180.0800.0000.102E  ;2225  NOP
U 102E. 0000.003D.0180.0800.0000.10DA  ;2226 =0 ERLOOP
U 102F. 0000.003C.0180.0800.0000.1030  ;2227  NOP
U 1030. 0000.003D.0180.0800.0000.1024  ;2228 =0 CALL[FPIRD]
U 1031. 0000.003C.0180.0800.0000.1032  ;2229  NOP
                ;2230
                ;2231 ; LOAD OPERANDS INTO THE FAD AR,BR LATCHES.
                ;2232 ; LOAD SIGNS AND EXPONENTS
                ;2233
                ;2234
U 1032. 0000.003D.0180.0800.0000.10E4  ;2235 =0 CALL[LDFAD]
U 1033. 0000.003C.0180.0800.0000.1034  ;2236  NOP
                ;2237
                ;2238 ; PERFORM FAD CONDITIONAL ADD/SUB, NORMALIZE AND UPDATE SIGN AND EXPONENT
                ;2239
U 1034. 0818.0039.4180.0800.0000.1119  ;2240 =0 D_K[.80],CALL[GENTRA]
U 1035. 0000.003C.5980.3C00.0000.114C  ;2241  ID[ACC.2]_D
U 114C. 0000.00FC.0380.0800.0000.114D  ;2242  TRAP.FPA
U 114D. 0000.003C.01D8.0800.0000.114E  ;2243  Q_ACC     ; FPA AT 080.
                ;2244       ; DO FAD CONDITIONAL ADD/SUB
                ;2245       ; TEST EXPONENTS. BEN/5
```

```
            ;2246       ; NORMALIZE
U 114E, 0000,003C,0180,0800,0000,114F  ;2247  NOP     ; FPA AT 052.
            ;2248       ; SUB*EXP. DIFF. <3
U 114F, 0000,003C,0180,0800,0000,1150  ;2249  NOP     ; FPA AT 058.
            ;2250       ; NR<63:52> CONTAIN A (1) ?
U 1150, 0000,003C,0180,0800,0000,1151  ;2251  NOP     ; FPA AT 059.
            ;2252       ; NR<63:52> CONTAIN A (1) ?
U 1151, 0000,003C,0180,0800,0000,1152  ;2253  NOP     ; FPA AT 05A.
            ;2254       ; NR<63:52> CONTAIN A (1) ?
U 1152, 0000,003C,59F0,2C00,0000,1153  ;2255  Q_ID[ACC.2]    ; FPA AT 05B . TEST FOR SWR.
            ;2256       ; NAD=05F IF OK.
            ;2257       ; NAD=05C IF NOT.
U 1153, 0001,203C,0180,09E8,0000,1154  ;2258  RC[0D]_Q    ; FPA AT 05C.
            ;2259       ; COULD NOT NORMALIZE IN (60)
            ;2260       ; LEFT SHIFTS OF NR.
            ;2261       ; FORCE ZEROS ON THE BUS.
U 1154, 0000,003C,0180,0800,0000,1156  ;2262  NOP     ; FPA AT 05F.
            ;2263       ; NORMALIZATION OK.
            ;2264       ; UP DATE SIGN AND EXPONENT.
            ;2265       ; FPA AT 056.
            ;2266       ; OUTPUT FRACTION HI.
            ;2267       ; SIGN, AND EXPONENT.
U 1156, 0000,003C,0180,0800,0000,1157  ;2268  NOP     ; FPA AT 057.
            ;2269       ; OUTPUT FRACTION LO
            ;2270       ;
U 1157, 0000,003C,0180,0800,0000,1158  ;2271  NOP     ; FPA AT 028.
            ;2272
            ;2273
            ;2274 ; TEST RESULTS
            ;2275
U 1158, 0010,0038,01C0,0968,0000,1159  ;2276  Q_RC[0D]    ; CHECK FPA STATUS REGISTER
U 1159, 0019,2034,81C0,0800,0000,115A  ;2277  Q_Q.AND.K[.3FF]    ; MASK NAD FIELD
U 115A, 0810,0038,0180,0970,0000,115B  ;2278  D_RC[0E]    ; EXPECTED NAD
U 115B, 001D,2000,0180,0800,0010,115C  ;2279  ALU_Q-D,CLK.UBCC
U 115C, 0000,013C,0180,0800,0000,1036  ;2280  Z?
U 1036, 0000,003D,0180,0800,0000,10DB  ;2281 =0 ERROR1
U 1037, 0000,003C,0180,0800,0000,1038  ;2282  NOP
```

```
;2283  .PAGE "TEST 288 NORMALIZER ROM TEST (1)"
;2284  ;*****************************************************************
;2285  ;**
;2286  ; NORMALIZER ROM TEST (1)
;2287  ;
;2288  ; TEST DESCRIPTION
;2289  ;
;2290  ; THIS TEST USES (96) OPERANDS TO TEST NORMALIZER ROM LOCATIONS
;2291  ; <88:AF>,<09:30>. THE DATA TABLE AT THE END OF THIS TEST CONTAINS
;2292  ; (48) OPERANDS. THE ADDITIONAL (48) ARE IDENTICAL TO THE DATA
;2293  ; TABLE EXCEPT THAT THE THE CONTENTS OF AR1,AND BR1, ARE SWAPPED.
;2294  ; THE TEST DOES ADDD R,R AND NORMALIZES AND ROUNDS THE RESULT.
;2295  ;
;2296  ; FPA UCODE USED
;2297  ;
;2298  ; LOCATION CONTENTS
;2299  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2300  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2301  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2302  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2303  ; 06D: BSC/R,SCR/CPU/FADC/BR,WAIT,NAD/083
;2304  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2305  ; 028: NAD/028
;2306  ; 080: FADC/A.B,BSC/FAL.X,NRC/LD,AMXC/COND,EALUC/A,EAC/LDPR,
;2307  ;  BEN/5,NAD/050
;2308  ; 050: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC,P.ADD,EALUC/A+B,AMXC/PR,
;2309  ;  BMXC/NK,NAD/056
;2310  ; 051: FADC/A.B,BSC/FAL.HL,NRC/LD,NAD/050
;2311  ; 052: NRC/SL,SWR?,NAD/058
;2312  ; 058: NRC/SL,SWR?,NAD/059
;2313  ; 059: NRC/SL,SWR?,NAD/05A
;2314  ; 05A: NRC/SL,SWR?,NAD/05B
;2315  ; 05B: NRC/SL,SWR?,NAD/05C
;2316  ; 05C: BSC/NH,SGNC/A.RES,EALUC/K3FF,WAIT,FPSYNC,NAD/054
;2317  ; 05F: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC/P.ADD,EALUC/A+B,AMXC/PR,
;2318  ;  BMXC/NK,NAD/056
;2319  ; 056: BMXC/NK,BSC/NH,AMXC/PR,EALUC/A,WAIT,FPSYNC,NAD/057
;2320  ; 057: BMXC/NK,BSC/NL,AMXC/PR,EALUC/A,NAD/028
;2321  ; 054: BSC/NH,SGNC/A.RES,EALUC/K3FF,NAD/028
;2322  ;
;2323  ; LOGIC DESCRIPTION
;2324  ;
;2325  ; THE NORMALIZER LOGIC IS ON THE FNM MODULE. THE NORMALIZER
;2326  ; INPUT TO UPDATE THE FINAL EXPONENT IS ON THE FCT MODULE.
;2327  ;
;2328  ; ERROR DESCRIPTION
;2329  ;
;2330  ; EXPECTED NORMALIZED RESULT HI
;2331  ; RECEIVED NORMALIZED RESULT HI
;2332  ; EXPECTED NORMALIZED RESULT LO
;2333  ; RECEIVED NORMALIZED RESULT LO
;2334  ; NORMALIZER ROM LOCATION BEING TESTED
;2335  ; ADDRESS OF SP1
;2336  ; ADDRESS OF SP2
;2337  ; ADDRESS OF RESULT
```

```
            ;2338 ; OPERATIONS COUNTER
            ;2339 ; COMPLEMENT FLAG
            ;2340 ;
            ;2341 ; SYNC POINT DESCRIPTION
            ;2342 ;
            ;2343 ;--
            ;2344 ;*****************************************************************
            ;2345 =0
U 1038. 0018.0039.F580.09F8.0000.10C8  ;2346 NROM: NEWTST[,A]
U 1039. 0018.0038.C580.09D0.0000.115D  ;2347  RC[0A]_K[.88]   ; ROM ADDRESS COUNTER
U 115D. 0818.0038.3180.0800.0000.115E  ;2348  D_K[.40]
U 115E. 0001.003C.0180.09C8.0000.115F  ;2349  RC[9]_D    ; ADDRESS OF FIRST SP1 OPERAND
U 115F. 0001.003C.0180.0A80.0000.1160  ;2350  R[0]_D
U 1160. 0819.0014.1180.0800.0000.1161  ;2351  ALU_D+K[.4],D_ALU
U 1161. 0001.003C.0180.09C0.0000.1162  ;2352  RC[8]_D    ; ADDRESS OF FIRST SP2 OPERAND
U 1162. 0001.003C.0180.0A88.0000.1163  ;2353  R[1]_D
U 1163. 0818.0038.7580.0800.0000.1164  ;2354  D_K[.20]
U 1164. 0000.003C.11F8.0800.0084.7165  ;2355  SC_K[.4],Q_0
U 1165. 0D00.003C.0180.0800.0000.1166  ;2356  D_DAL.SC
U 1166. 0001.003C.0180.09B8.0000.1167  ;2357  RC[7]_D    ; ADDRESS OF FIRST RESULT
U 1167. 0001.003C.0180.0A90.0000.1168  ;2358  R[2]_D
U 1168. 0003.003C.0180.09A8.0000.1169  ;2359  RC[5]_0    ; INITIALIZE COMPLEMENT FLAG
U 1169. 0018.0038.2D80.09B0.0000.116A  ;2360  RC[6]_K[.28]   ; OPERATIONS COUNTER SET TO 48
U 116A. 0818.0038.4D80.0800.0000.116B  ;2361  D_K[.FF00]
U 116B. 0819.0030.4180.0800.0000.116C  ;2362  D_D.OR.K[.80]
U 116C. 0001.0028.0180.0AA0.0000.103A  ;2363  R[4]_NOT.D   ; MASK FOR BITS <15:07>
U 103A. 0000.003D.0180.0800.0000.10DA  ;2364 =0 ERLOOP
U 103B. 0000.003C.0180.0800.0000.103C  ;2365  NOP
            ;2366 =0
U 103C. 0000.003D.0180.0800.0000.1114  ;2367 NROM1: CALL[FPINIT]
U 103D. 0000.003C.0180.0800.0000.116D  ;2368  NOP
U 116D. 0018.0038.7580.0800.0200.103E  ;2369  VA_K[.20]
U 103E. 0000.003D.0180.0800.0000.111F  ;2370 =0 CALL[LOADIB]   ; ADDD R,R
U 103F. 0010.0038.0180.0948.0200.116E  ;2371  VA_RC[9]
U 116E. 0000.003C.0180.4000.0000.116F  ;2372  D[LONG]_CACHE    ; FETCH SP1 HI
U 116F. 0001.003C.0180.0AA8.0000.1170  ;2373  R[5]_D    ; SAVE TO LOAD SA,LA.
U 1170. 0000.003C.01C0.0A20.0000.1171  ;2374  Q_R[4]
U 1171. 001D.0034.0180.0AB0.0000.1172  ;2375  ALU_D.AND.Q,R[6]_ALU  ; MASK OUT SIGN AND EXPONENT
            ;2376       ; SAVE FOR FAD AR1
U 1172. 0003.003C.0180.0AB8.0000.1173  ;2377  R[7]_0    ; SAVE FOR FAD AR0 (SP1 LO)
U 1173. 0010.0038.0180.0940.0200.1174  ;2378  VA_RC[8]
U 1174. 0000.003C.0180.4000.0000.1175  ;2379  D[LONG]_CACHE    ; FETCH SP2 HI
U 1175. 0001.003C.0180.0A98.0000.1176  ;2380  R[3]_D    ; SAVE TO LOAD SB,LB
U 1176. 001D.0034.0180.0AC0.0000.1177  ;2381  ALU_D.AND.Q,R[8]_ALU  ; MASK OUT SIGN AND EXPONENT
            ;2382       ; SAVE FOR FAD BR1
U 1177. 0003.003C.0180.0AC8.0000.1178  ;2383  R[9]_0    ; SAVE FOR FAD BR0 (SP2 LO)
U 1178. 0010.0038.0180.0938.0200.1179  ;2384  VA_RC[7]
U 1179. 0000.003C.0180.4000.0000.117A  ;2385  D[LONG]_CACHE
U 117A. 0001.003C.0180.09F0.0000.117B  ;2386  RC[0E]_D   ; SAVE EXPECTED RESULT HI
U 117B. 0003.003C.0180.09E0.0000.117C  ;2387  RC[0C]_0   ; SAVE EXPECTED RESULT LO
            ;2388
            ;2389 ; CHECK SWAP  FLAG
            ;2390
U 117C. 0010.0038.0180.0928.0010.117D  ;2391 CMP1: ALU_RC[5],CLK.UBCC
U 117D. 0000.013C.0180.0800.0000.1040  ;2392  Z?
```

```
U 1040. 0000.003C.0180.0800.0000.117E  ;2393 =0 J/SWP1
U 1041. 0000.003C.0180.0800.0000.1042  ;2394  J/NROM2    ; FLAG IS NOT SET
U 117E. 0800.003C.0180.0A30.0000.117F  ;2395 SWP1: D_R[6]
U 117F. 0000.003C.01C0.0A40.0000.1180  ;2396  Q_R[8]
U 1180. 0001.203C.0180.0AB0.0000.1181  ;2397  R[6]_Q
U 1181. 0001.003C.0180.0AC0.0000.1182  ;2398  R[8]_D    ; SWAP AR1,BR1
U 1182. 0800.003C.0180.0A28.0000.1183  ;2399  D_R[5]
U 1183. 0000.003C.01C0.0A18.0000.1184  ;2400  Q_R[3]
U 1184. 0001.203C.0180.0AA8.0000.1185  ;2401  R[5]_Q
U 1185. 0001.003C.0180.0A98.0000.1042  ;2402  R[3]_D    ; SWAP AR1,BR1 SIGNS AND EXPONENTS.
          ;2403
          ;2404
          ;2405 ; LOAD OPERANDS INTO THE FAD AR,BR LATCHES.
          ;2406 ; LOAD SIGNS AND EXPONENTS
          ;2407
          ;2408
          ;2409 =0
U 1042. 0000.003D.0180.0800.0000.10E4  ;2410 NROM2: CALL[LDFAD]
U 1043. 0000.003C.0180.0800.0000.1044  ;2411  NOP
          ;2412
          ;2413 ; PERFORM FAD CONDITIONAL ADD/SUB, NORMALIZE AND UPDATE SIGN AND EXPONENT
          ;2414
U 1044. 0000.003D.0180.0800.0000.101A  ;2415 =0 CALL[NORM]
U 1045. 0000.003C.0180.0800.0000.1186  ;2416  NOP
          ;2417
          ;2418
          ;2419 ; TEST RESULTS
          ;2420 ; NOTE: SUBROUTINE NORM SAVES RESULT HI IN RC(D), RESULT LO IN RC(B).
          ;2421
U 1186. 0810.0038.0180.0970.0000.1187  ;2422  D_RC[0E]   ; FETCH EXPECTED RESULT HI
U 1187. 0010.0038.01C0.0968.0000.1188  ;2423  Q_RC[0D]   ; RECEIVED RESULT HI
U 1188. 001D.0000.0180.0800.0010.1189  ;2424  ALU_D-Q,CLK.UBCC  ; CHECK RESULT HI
U 1189. 0000.013C.0180.0800.0000.1046  ;2425  Z?
U 1046. 0000.003D.0180.0800.0000.10DB  ;2426 =0 ERROR1
U 1047. 0810.0038.0180.0960.0000.118A  ;2427  D_RC[0C]   ; FETCH EXPECTED RESULT LO
U 118A. 0010.0038.01C0.0958.0000.118B  ;2428  Q_RC[0B]   ; RECEIVED RESULT LO
U 118B. 001D.0000.0180.0800.0010.118C  ;2429  ALU_D-Q,CLK.UBCC  ; CHECK RESULT LO
U 118C. 0000.013C.0180.0800.0000.1048  ;2430  Z?
U 1048. 0000.003D.0180.0800.0000.10DB  ;2431 =0 ERROR1
          ;2432
          ;2433 ; TEST OPERATIONS COUNTER
          ;2434
U 1049. 0810.0038.0180.0930.0000.118D  ;2435  D_RC[6]
U 118D. 0019.0000.0580.09B0.0010.118E  ;2436  ALU_D-K[.1],RC[6]_ALU,CLK.UBCC
U 118E. 0000.013C.0180.0800.0000.104A  ;2437  Z?
U 104A. 0000.003C.0180.0800.0000.1190  ;2438 =0 J/SWP2
          ;2439
          ;2440 ; CHECK SWAP FLAG. IF SET, 96 OPERATIONS HAVE BEEN DONE
          ;2441 ; (48) WITH AR1=SP1 HI, BR1=SP2 HI AND (48) WITH
          ;2442 ; AR1=SP2 HI, BR1=SP1 HI.
          ;2443
U 104B. 0010.0038.0180.0928.0010.118F  ;2444  ALU_RC[5],CLK.UBCC
U 118F. 0000.013C.0180.0800.0000.104C  ;2445  Z?
U 104C. 0000.003C.0180.0800.0000.11A1  ;2446 =0 J/NROM4
U 104D. 0000.003C.0180.0800.0000.11AB  ;2447  J/ENTST1
```

```
            :2448
            :2449 ; UPDATE OPERAND AND RESULT INDEX REGISTERS.
            :2450
U 1190. 0010.0038.0180.0928.0010.1191  :2451 SWP2: ALU_RC[5],CLK.UBCC
U 1191. 0000.013C.0180.0800.0000.1050  :2452  Z?
U 1050. 0000.003C.0180.0800.0000.1192  :2453 =0 J/LOROM
U 1051. 0000.003C.0180.0800.0000.1199  :2454  J/HIROM
U 1192. 0810.0038.0180.0950.0000.1193  :2455 LOROM: D_RC[0A]    ; FETCH ROM ADDRESS COUNTER
U 1193. 0019.0034.CDC0.0800.0000.1194  :2456  ALU_D.AND.K[.F0],Q_ALU  ; SAVE HI NIBBLE
U 1194. 0819.0034.6180.0800.0000.1195  :2457  ALU_D.AND.K[.F],D_ALU  ; SAVE LO NIBBLE
U 1195. 0819.0000.0180.0800.0010.1196  :2458  ALU_D-K[.8],D_ALU,CLK.UBCC ; SUBTRACT (8) FTOM LO NIBBLE
U 1196. 0000.013C.5D80.0800.0000.1052  :2459  Z?,KMX/.7
U 1052. 0000.003C.0180.0800.0000.1197  :2460 =0 J/LO1
U 1053. 0019.2000.5DC0.0800.0000.1197  :2461  ALU_Q-K[.7],Q_ALU  ; SET HI NIBBLE=1
U 1197. 0019.2000.01C0.0800.0000.1198  :2462 LO1: ALU_Q-K[.8],Q_ALU  ; SUBTRACT (8) FROM HI NIBBLE
U 1198. 001D.0030.0180.09D0.0000.119B  :2463  ALU_D.OR.Q,RC[0A]_ALU,J/NROM3 ; COMBINE HI AND LO NIBBLES
            :2464       ; SAVE LO ROM ADDRESS
U 1199. 0810.0038.0180.0950.0000.119A  :2465 HIROM: D_RC[0A]
U 119A. 0019.0014.0580.09D0.0000.119B  :2466  ALU_D+K[.1],RC[0A]_ALU  ; UPDATE ROM ADDRESS COUNTER
U 119B. 0810.0038.0180.0948.0000.119C  :2467 NROM3: D_RC[9]
U 119C. 0019.0014.0180.09C8.0000.119D  :2468  ALU_D+K[.8],RC[9]_ALU  ; UPDATE TO NEXT SP1
U 119D. 0810.0038.0180.0940.0000.119E  :2469  D_RC[8]
U 119E. 0019.0014.0180.09C0.0000.119F  :2470  ALU_D+K[.8],RC[8]_ALU  ; UPDATE TO NEXT SP2
U 119F. 0810.0038.0180.0938.0000.11A0  :2471  D_RC[7]
U 11A0. 0019.0014.1180.09B8.0000.103C  :2472  ALU_D+K[.4],RC[7]_ALU,J/NROM1 ; UPDATE TO NEXT RESULT
            :2473
            :2474 ; SET SWAP FLAG AND RE-INITIALIZE OPERAND AND RESULT INDEX
            :2475 ; REGISTERS FOR THE NEXT (48) OPERATIONS.
            :2476
U 11A1. 0810.0038.0180.0928.0000.11A2  :2477 NROM4: D_RC[5]
U 11A2. 0019.0030.0580.09A8.0000.11A3  :2478  ALU_D.OR.K[.1],RC[5]_ALU ; SET COMPLEMENT FLAG
U 11A3. 0003.003C.0180.09D0.0000.11A4  :2479  RC[0A]_D    ; UPDATE ROM ADDRESS COUNTER
U 11A4. 0018.0038.2D80.09B0.0000.11A5  :2480  RC[6]_K[.28]    ; INIT. OPERATIONS COUNTER TO 48
U 11A5. 0800.003C.0180.0A00.0000.11A6  :2481  D_R[0]
U 11A6. 0001.003C.0180.09C8.0000.11A7  :2482  RC[9]_D     ; RE_INITIALIZE TO FIRST SP1
U 11A7. 0800.003C.0180.0A08.0000.11A8  :2483  D_R[1]
U 11A8. 0001.003C.0180.09C0.0000.11A9  :2484  RC[8]_D     ; RE-INITIALIZE TO FIRST SP2
U 11A9. 0800.003C.0180.0A10.0000.11AA  :2485  D_R[2]
U 11AA. 0001.003C.0180.09B8.0000.103C  :2486  RC[7]_D,J/NROM1    ; RE-INITIALIZE TO FIRST RESULT
U 11AB. 0000.003C.0180.0800.0000.1054  :2487 ENTST1: NOP
            :2488
```

```
;2489  .PAGE "TEST 289 NORMALIZER ROM TEST (2)"
;2490 ;*************************************************************************
;2491 ;**
;2492 ; NORMALIZER ROM TEST (2)
;2493 ;
;2494 ; TEST DESCRIPTION
;2495 ;
;2496 ; THIS TEST USES (128) OPERANDS TO TEST NORMALIZER ROM LOCATIONS
;2497 ; <B0:DF>,<38:70>. THE DATA TABLE AT THE END OF THIS TEST CONTAINS
;2498 ; (64) OPERANDS. THE ADDITIONAL (64) ARE IDENTICAL TO THE DATA
;2499 ;  TABLE EXCEPT THAT THE THE CONTENTS OF AR1 AND BR1 ARE SWAPPED.
;2500 ; THE RESULTS OF THE (64) OPERATIONSIS A REPETETIVE PATTERN OF (8)
;2501 ; QUAD WORDS. FOR EXAMPLE, THE FIRST (8) OPERATIONS YIELD THE (8)
;2502 ; RESULTS IN THE TABLE. THE THE NEXT (8) OPERATIONS YIELD THE SAME
;2503 ; (8) RESULT PATTERNS. THIS SEQUENCE IS REPEATED UNTILL (64)
;2504 ; OPERATIONS ARE COMPLETED. THE NEXT (64) OPERATIONS REPEAT THE ABOVE
;2505 ; MENTIONED SEQUENCE.
;2506 ;
;2507 ; FPA UCODE USED
;2508 ;
;2509 ; LOCATION CONTENTS
;2510 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2511 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2512 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2513 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2514 ; 06D: BSC/R,SCR/CPU/FADC/BR,WAIT,NAD/083
;2515 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2516 ; 028: NAD/028
;2517 ; 080: FADC/A.B,BSC/FAL.X,NRC/LD,AMXC/COND,EALUC/A,EAC/LDPR,
;2518 ;  BEN/S,NAD/050
;2519 ; 050: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC,P.ADD,EALUC/A+B,AMXC/PR,
;2520 ;  BMXC/NK,NAD/056
;2521 ; 051: FADC/A.B,BSC/FAL.HL,NRC/LD,NAD/050
;2522 ; 052: NRC/SL,SWR?,NAD/058
;2523 ; 058: NRC/SL,SWR?,NAD/059
;2524 ; 059: NRC/SL,SWR?,NAD/05A
;2525 ; 05A: NRC/SL,SWR?,NAD/05B
;2526 ; 05B: NRC/SL,SWR?,NAD/05C
;2527 ; 05C: BSC/NH,SGNC/A.RES,EALUC/K3FF,WAIT,FPSYNC,NAD/054
;2528 ; 05F: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC/P.ADD,EALUC/A+B,AMXC/PR,
;2529 ;  BMXC/NK,NAD/056
;2530 ; 056: BMXC/NK,BSC/NH,AMXC/PR,EALUC/A,WAIT,FPSYNC,NAD/057
;2531 ; 057: BMXC/NK,BSC/NL,AMXC/PR,EALUC/A,NAD/028
;2532 ; 054: BSC/NH,SGNC/A.RES,EALUC/K3FF,NAD/028
;2533 ;
;2534 ; LOGIC DESCRIPTION
;2535 ;
;2536 ; THE NORMALIZER LOGIC IS ON THE FNM MODULE. THE NORMALIZER
;2537 ; INPUT TO UPDATE THE FINAL EXPONENT IS ON THE FCT MODULE.
;2538 ;
;2539 ; ERROR DESCRIPTION
;2540 ;
;2541 ; EXPECTED NORMALIZED RESULT HI
;2542 ; RECEIVED NORMALIZED RESULT HI
;2543 ; EXPECTED NORMALIZED RESULT LO
```

```
            ;2544 ; RECEIVED NORMALIZED RESULT LO
            ;2545 ; NORMALIZER ROM LOCATION BEING TESTED
            ;2546 ; ADDRESS OF SP1
            ;2547 ; ADDRESS OF SP2
            ;2548 ; ADDRESS OF RESULT
            ;2549 ; OPERATIONS COUNTER
            ;2550 ; COMPLEMENT FLAG
            ;2551 ;
            ;2552 ; SYNC POINT DESCRIPTION
            ;2553 ;
            ;2554 ;--
            ;2555 ;*************************************************************************
            ;2556
            ;2557
            ;2558 =0
U 1054, 0018.0039.F580.09F8.0000.10C8  ;2559 NROM6: NEWTST[.A]
U 1055, 0818.0038.9580.0800.0000.11AC  ;2560  D_K[.B0]
U 11AC, 0019.0014.0180.09D0.0000.11AD  ;2561  ALU_D+K[.8],RC[0A]_ALU  ; ROM ADDRESS COUNTER
U 11AD, 0818.0038.7980.0800.0000.11AE  ;2562  D_K[.30]
U 11AE, 0000.003C.11F8.0800.0084.71AF  ;2563  SC_K[.4],Q_0
U 11AF, 0D00.003C.0180.0800.0000.11B0  ;2564  D_DAL.SC
U 11B0, 0001.003C.0180.09C8.0000.11B1  ;2565  RC[9]_D     ; ADDRESS OF FIRST SP1 OPERAND
U 11B1, 0001.003C.0180.0A80.0000.11B2  ;2566  R[0]_D
U 11B2, 0819.0014.0180.0800.0000.11B3  ;2567  ALU_D+K[.8],D_ALU
U 11B3, 0001.003C.0180.09C0.0000.11B4  ;2568  RC[8]_D     ; ADDRESS OF FIRST SP2 OPERAND
U 11B4, 0001.003C.0180.0A88.0000.11B5  ;2569  R[1]_D
U 11B5, 0818.0038.0180.0800.0000.11B6  ;2570  D_K[.8]
U 11B6, 0000.003C.01F8.0800.0084.71B7  ;2571  SC_K[.8],Q_0
U 11B7, 0D00.003C.0180.0800.0000.11B8  ;2572  D_DAL.SC
U 11B8, 0001.003C.0180.09B8.0000.11B9  ;2573  RC[7]_D     ; ADDRESS OF FIRST RESULT
U 11B9, 0001.003C.0180.0A90.0000.11BA  ;2574  R[2]_D
U 11BA, 0003.003C.0180.09A8.0000.11BB  ;2575  RC[5]_0     ; INITIALIZE COMPLEMENT FLAG
U 11BB, 0018.0038.3180.09B0.0000.11BC  ;2576  RC[6]_K[.40]    ; OPERATIONS COUNTER SET TO 64
U 11BC, 0818.0038.4D80.0800.0000.11BD  ;2577  D_K[.FF00]
U 11BD, 0819.0030.4180.0800.0000.11BE  ;2578  D_D.OR.K[.80]
U 11BE, 0001.0028.0180.0AA0.0000.11BF  ;2579  R[4]_NOT.D    ; MASK FOR BITS <15:07>
U 11BF, 0003.003C.0180.0AE0.0000.1056  ;2580  R[0C]_0     ; SET RESULT COUNTER
U 1056, 0000.003D.0180.0800.0000.10DA  ;2581 =0 ERLOOP
U 1057, 0000.003C.0180.0800.0000.1058  ;2582  NOP
            ;2583 =0
U 1058, 0000.003D.0180.0B00.0000.1114  ;2584 NROM7: CALL[FPINIT]
U 1059, 0000.003C.0180.0B00.0000.11C0  ;2585  NOP
U 11C0, 0018.0038.7580.0B00.0200.105C  ;2586  VA_K[.20]
U 105C, 0000.003D.0180.0B00.0000.111F  ;2587 =0 CALL[LOADIB]    ; ADDD R,R
U 105D, 0010.0038.0180.0948.0200.11C1  ;2588  VA_RC[9]
U 11C1, 0000.003C.0180.4000.0000.11C2  ;2589  D[LONG]_CACHE    ; FETCH SP1 HI
U 11C2, 0001.003C.0180.0AA8.0000.11C3  ;2590  R[5]_D     ; SAVE TO LOAD SA.LA.
U 11C3, 0000.003C.01C0.0A20.0000.11C4  ;2591  Q_R[4]     ; MASK
U 11C4, 001D.0034.0180.0AB0.0000.11C5  ;2592  ALU_D.AND.Q,R[6]_ALU  ; MASK OUT SIGN AND EXPONENT
            ;2593       ; SAVE FOR FAD AR1
U 11C5, 0810.0038.0180.0948.0000.11C6  ;2594  D_RC[9]
U 11C6, 0019.0014.1180.0B00.0200.11C7  ;2595  ALU_D+K[.4],VA_ALU
U 11C7, 0000.003C.0180.4000.0000.11C8  ;2596  D[LONG]_CACHE    ; FETCH SP1 LO
U 11C8, 0001.003C.0180.0AB8.0000.11C9  ;2597  R[7]_D     ; SAVE FOR FAD AR0
U 11C9, 0010.0038.0180.0940.0200.11CA  ;2598  VA_RC[8]
```

```
U 11CA. 0000.003C.0180.4000.0000.11CB  ;2599  D[LONG]_CACHE    ; FETCH SP2 HI
U 11CB. 0001.003C.0180.0A98.0000.11CC  ;2600  R[3]_D     ; SAVE TO LOAD SB.LB
U 11CC. 0000.003C.01C0.0A20.0000.11CD  ;2601  Q_R[4]     ; MASK
U 11CD. 001D.0034.0180.0AC0.0000.11CE  ;2602  ALU_D.AND.Q,R[8]_ALU   ; MASK OUT SIGN AND EXPONENT
          ;2603       ; SAVE FOR FAD BR1
U 11CE. 0810.0038.0180.0940.0000.11CF  ;2604  D_RC[8]
U 11CF. 0019.0014.1180.0800.0200.11D0  ;2605  ALU_D+K[.4],VA_ALU
U 11D0. 0000.003C.0180.4000.0000.11D1  ;2606  D[LONG]_CACHE    ; FETCH SP2 LO
U 11D1. 0001.003C.0180.0AC8.0000.11D2  ;2607  R[9]_D     ; SAVE FOR FAD BR0
          ;2608
          ;2609 ; INCREMENT AND CHECK RESULT COUNTER.
          ;2610
U 11D2. 0010.0038.0180.0938.0200.11D3  ;2611  VA_RC[7]
U 11D3. 0000.003C.0180.4000.0000.11D4  ;2612  D[LONG]_CACHE
U 11D4. 0001.003C.0180.09F0.0000.11D5  ;2613  RC[0E]_D    ; SAVE EXPECTED RESULT HI
U 11D5. 0003.003C.0180.09E0.0000.11D6  ;2614  RC[0C]_0    ; SAVE EXPECTED RESULT LO
U 11D6. 0018.0038.01C0.0800.0000.11D7  ;2615  Q_K[.8]
U 11D7. 0800.003C.0180.0A60.0000.11D8  ;2616  D_R[0C]
U 11D8. 001D.2000.0180.0800.0010.11D9  ;2617  ALU_Q-D.CLK.UBCC
U 11D9. 0000.013C.0180.0800.0000.1060  ;2618  Z?
U 1060. 0000.003C.0180.0800.0000.11DF  ;2619 =0 J/NXT
U 1061. 0800.003C.0180.0A10.0000.11DA  ;2620  D_R[2]     ; RE-INITIALIZE TO FIRST RESULT
U 11DA. 0018.0038.0580.0AE0.0000.11DB  ;2621  R[0C]_K[.1]    ; RE-INITIALIZE RESULT COUNTER
U 11DB. 0001.003C.0180.09B8.0000.11DC  ;2622  RC[7]_D
U 11DC. 0010.0038.0180.0938.0200.11DD  ;2623  VA_RC[7]
U 11DD. 0000.003C.0180.4000.0000.11DE  ;2624  D[LONG]_CACHE
U 11DE. 0001.003C.0180.09F0.0000.11E1  ;2625  RC[0E]_D,J/NXT1    ; SAVE EXPECTED
U 11DF. 0818.0014.0580.0A60.0000.11E0  ;2626 NXT: ALU_R[0C]+K[.1],D_ALU
U 11E0. 0001.003C.0180.0AE0.0000.11E1  ;2627  R[0C]_D
          ;2628
          ;2629 ; CHECK SWAP FLAG
          ;2630
U 11E1. 0010.0038.0180.0928.0010.11E2  ;2631 NXT1: ALU_RC[5],CLK.UBCC
U 11E2. 0000.013C.0180.0800.0000.1062  ;2632  Z?
U 1062. 0000.003C.0180.0800.0000.11E3  ;2633 =0 J/SWP3
U 1063. 0000.003C.0180.0800.0000.1064  ;2634  J/NROM8     ; FLAG IS NOT SET
U 11E3. 0800.003C.0180.0A30.0000.11E4  ;2635 SWP3: D_R[6]
U 11E4. 0000.003C.01C0.0A40.0000.11E5  ;2636  Q_R[8]
U 11E5. 0001.203C.0180.0AB0.0000.11E6  ;2637  R[6]_Q
U 11E6. 0001.003C.0180.0AC0.0000.11E7  ;2638  R[8]_D     ; SWAP AR1,BR1
U 11E7. 0800.003C.0180.0A38.0000.11E8  ;2639  D_R[7]
U 11E8. 0000.003C.01C0.0A48.0000.11E9  ;2640  Q_R[9]
U 11E9. 0001.203C.0180.0AB8.0000.11EA  ;2641  R[7]_Q
U 11EA. 0001.003C.0180.0AC8.0000.11EB  ;2642  R[9]_D     ; SWAP AR0,BR0
U 11EB. 0800.003C.0180.0A28.0000.11EC  ;2643  D_R[5]
U 11EC. 0000.003C.01C0.0A18.0000.11ED  ;2644  Q_R[3]
U 11ED. 0001.203C.0180.0AA8.0000.11EE  ;2645  R[5]_Q
U 11EE. 0001.003C.0180.0A98.0000.1064  ;2646  R[3]_D     ; SWAP AR1,BR1, SIGNS AND EXPONENTS.
          ;2647
          ;2648 ; LOAD OPERANDS INTO THE FAD AR,BR LATCHES.
          ;2649 ; LOAD SIGNS AND EXPONENTS
          ;2650
          ;2651 =0
U 1064. 0000.003D.0180.0800.0000.10E4  ;2652 NROM8: CALL[LDFAD]
U 1065. 0000.003C.0180.0800.0000.1066  ;2653  NOP
```

```
            ;2654
            ;2655 ; PERFORM FAD CONDITIONAL ADD/SUB, NORMALIZE AND UPDATE SIGN AND EXPONENT
            ;2656
U 1066, 0000.003D.0180.0800.0000.101A  ;2657 =0 CALL[NORM]
U 1067, 0000.003C.0180.0800.0000.11EF  ;2658  NOP
            ;2659
            ;2660
            ;2661 ; TEST RESULTS
            ;2662 ; NOTE: SUBROUTINE NORM SAVES RESULT HI IN RC(D), RESULT LO IN RC(B).
            ;2663
U 11EF, 0810.0038.0180.0970.0000.11F0  ;2664  D_RC[0E]    ; FETCH EXPECTED RESULT HI
U 11F0, 0010.0038.01C0.0968.0000.11F1  ;2665  Q_RC[0D]    ; RECEIVED RESULT HI
U 11F1, 001D.0000.0180.0800.0010.11F2  ;2666  ALU_D-Q,CLK.UBCC  ; CHECK RESULT HI
U 11F2, 0000.013C.0180.0800.0000.1068  ;2667  Z?
U 1068, 0000.003D.0180.0800.0000.10DB  ;2668 =0 ERROR1
U 1069, 0810.0038.0180.0960.0000.11F3  ;2669  D_RC[0C]    ; FETCH EXPECTED RESULT LO
U 11F3, 0010.0038.01C0.0958.0000.11F4  ;2670  Q_RC[0B]    ; RECEIVED RESULT LO
U 11F4, 001D.0000.0180.0800.0010.11F5  ;2671  ALU_D-Q,CLK.UBCC  ; CHECK RESULT LO
U 11F5, 0000.013C.0180.0800.0000.106A  ;2672  Z?
U 106A, 0000.003D.0180.0800.0000.10DB  ;2673 =0 ERROR1
            ;2674
            ;2675 ; TEST OPERATIONS COUNTER
            ;2676
U 106B, 0810.0038.0180.0930.0000.11F6  ;2677  D_RC[6]
U 11F6, 0019.0000.0580.09B0.0010.11F7  ;2678  ALU_D-K[.1],RC[6]_ALU,CLK.UBCC
U 11F7, 0000.013C.0180.0800.0000.106C  ;2679  Z?
U 106C, 0000.003C.0180.0800.0000.11F9  ;2680 =0 J/SWP4
            ;2681
            ;2682 ; CHECK SWAP FLAG. IF SET, 128 OPERATIONS HAVE BEEN DONE
            ;2683 ; (64) WITH OPERANDS AND RESULT SIGNS AS SPECIFIED IN THE TABLE.
            ;2684 ; (64) WITH SP1 HI AND LO SWAPPED WITH SP2 HI AND LO.
            ;2685
U 106D, 0010.0038.0180.0928.0010.11F8  ;2686  ALU_RC[5],CLK.UBCC
U 11F8, 0000.013C.0180.0800.0000.106E  ;2687  Z?
U 106E, 0000.003C.0180.0800.0000.120A  ;2688 =0 J/NROM10
U 106F, 0000.003C.0180.0800.0000.1214  ;2689  J/ENTST2
            ;2690
            ;2691 ; UPDATE OPERAND AND RESULT INDEX REGISTERS.
            ;2692
U 11F9, 0010.0038.0180.0928.0010.11FA  ;2693 SWP4: ALU_RC[5],CLK.UBCC
U 11FA, 0000.013C.0180.0800.0000.1070  ;2694  Z?
U 1070, 0000.003C.0180.0800.0000.11FB  ;2695 =0 J/LOROM1
U 1071, 0000.003C.0180.0800.0000.1202  ;2696  J/HIROM1
U 11FB, 0810.0038.0180.0950.0000.11FC  ;2697 LOROM1: D_RC[0A]    ; FETCH ROM ADDRESS COUNTER
U 11FC, 0019.0034.CDC0.0800.0000.11FD  ;2698  ALU_D.AND.K[.F0],Q_ALU  ; SAVE HI NIBBLE
U 11FD, 0819.0034.6180.0800.0000.11FE  ;2699  ALU_D.AND.K[.F],D_ALU  ; SAVE LO NIBBLE
U 11FE, 0819.0000.0180.0800.0010.11FF  ;2700  ALU_D-K[.8],D_ALU,CLK.UBCC ; SUBTRACT (8) FROM LO NIBBLE
U 11FF, 0000.013C.5D80.0800.0000.1072  ;2701  Z?,KMX/.7
U 1072, 0000.003C.0180.0800.0000.1200  ;2702 =0 J/LO2
U 1073, 0019.2000.5DC0.0800.0000.1200  ;2703  ALU_Q-K[.7],Q_ALU  ; SET HI NIBBLE TO (1)
U 1200, 0019.2000.01C0.0800.0000.1201  ;2704 LO2: ALU_Q-K[.8],Q_ALU  ; SUBTRACT (8) FROM HI NIBBLE.
U 1201, 001D.0030.0180.09D0.0000.1204  ;2705  ALU_D.OR.Q,RC[0A]_ALU,J/NROM9 ; COMBINE HI AND LO NIBBLES
            ;2706       ; SAVE LO ROM ADDRESS
U 1202, 0810.0038.0180.0950.0000.1203  ;2707 HIROM1: D_RC[0A]
U 1203, 0019.0014.0580.09D0.0000.1204  ;2708  ALU_D+K[.1],RC[0A]_ALU  ; UPDATE ROM ADDRESS COUNTER
```

```
U 1204, 0810.0038.0180.0948.0000.1205  ;2709 NROM9: D_RC[9]
U 1205, 0019.0014.6580.09C8.0000.1206  ;2710  ALU_D+K[.10],RC[9]_ALU  ; UPDATE TO NEXT SP1
U 1206, 0810.0038.0180.0940.0000.1207  ;2711  D_RC[8]
U 1207, 0019.0014.6580.09C0.0000.1208  ;2712  ALU_D+K[.10],RC[8]_ALU  ; UPDATE TO NEXT SP2
U 1208, 0810.0038.0180.0938.0000.1209  ;2713  D_RC[7]
U 1209, 0019.0014.1180.09B8.0000.1058  ;2714  ALU_D+K[.4],RC[7]_ALU,J/NROM7 ; UPDATE TO NEXT RESULT
          ;2715
          ;2716 ; SET SWAP FLAG AND RE-INITIALIZE OPERAND AND RESULT INDEX
          ;2717 ; REGISTERS FOR THE NEXT (64) OPERATIONS.
          ;2718
U 120A, 0810.0038.0180.0928.0000.120B  ;2719 NROM10: D_RC[5]
U 120B, 0019.0030.0580.09A8.0000.120C  ;2720  ALU_D.OR.K[.1],RC[5]_ALU ; SET COMPLEMENT FLAG
U 120C, 0018.0038.9580.09D0.0000.120D  ;2721  RC[0A]_K[.B0]   ; UPDATE ROM ADDRESS COUNTER
U 120D, 0018.0038.3180.09B0.0000.120E  ;2722  RC[6]_K[.40]   ; INIT. OPERATIONS COUNTER TO 48
U 120E, 0800.003C.0180.0A00.0000.120F  ;2723  D_R[0]
U 120F, 0800.003C.0180.0A08.0000.1210  ;2724  D_R[1]
U 1210, 0001.003C.0180.09C0.0000.1211  ;2725  RC[8]_D    ; RE-INITIALIZE TO FIRST SP2
U 1211, 0800.003C.0180.0A10.0000.1212  ;2726  D_R[2]
U 1212, 0018.0038.0580.0A98.0000.1213  ;2727  R[3]_K[.1]
U 1213, 0001.003C.0180.09B8.0000.1058  ;2728  RC[7]_D,J/NROM7   ; RE-INITIALIZE TO FIRST RESULT
          ;2729
          ;2730
U 1214, 0000.003C.0180.0800.0000.1074  ;2731 ENTST2: NOP
          ;2732
          ;2733
```

```
;2734   .PAGE "TEST 28A TEST EXPONENT OVERFLOW DUE TO ROUNDING"
;2735  ;*****************************************************************
;2736  ;**
;2737  ; TEST EXPONENT OVERFLOW DUE TO ROUNDING
;2738  ;
;2739  ; TEST DESCRIPTION
;2740  ;
;2741  ; THIS TEST LOADS THE FPA WITH OPERANDS THAT WILL CAUSE
;2742  ; EXPONENT OVERFLOW DUE TO ROUNDING.
;2743  ;
;2744  ; FPA UCODE USED
;2745  ; LOCATION CONTENTS
;2746  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2747  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2748  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2749  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2750  ; 06D: BSC/R,SCR/CPU/FADC/BR,WAIT,NAD/083
;2751  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2752  ; 028: NAD/028
;2753  ; 080: FADC/A.B,BSC/FAL.X,NRC/LD,AMXC/COND,EALUC/A,EAC/LDPR,
;2754  ;  BEN/5,NAD/050
;2755  ; 050: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC,P.ADD,EALUC/A+B,AMXC/PR,
;2756  ;  BMXC/NK,NAD/056
;2757  ; 051: FADC/A.B,BSC/FAL.HL,NRC/LD,NAD/050
;2758  ; 052: NRC/SL,SWR?,NAD/058
;2759  ; 058: NRC/SL,SWR?,NAD/059
;2760  ; 059: NRC/SL,SWR?,NAD/05A
;2761  ; 05A: NRC/SL,SWR?,NAD/05B
;2762  ; 05B: NRC/SL,SWR?,NAD/05C
;2763  ; 05C: BSC/NH,SGNC/A.RES,EALUC/K3FF,WAIT,FPSYNC,NAD/054
;2764  ; 05F: EAC/LDPR,SGNC/A.RES,BSC/NH,MSC/P.ADD,EALUC/A+B,AMXC/PR,
;2765  ;  BMXC/NK,NAD/056
;2766  ; 056: BMXC/NK,BSC/NH,AMXC/PR,EALUC/A,WAIT,FPSYNC,NAD/057
;2767  ; 057: BMXC/NK,BSC/NL,AMXC/PR,EALUC/A,NAD/028
;2768  ; 054: BSC/NH,SGNC/A.RES,EALUC/K3FF,NAD/028
;2769  ;
;2770  ; LOGIC DESCRIPTION
;2771  ;
;2772  ; ERROR DESCRIPTION
;2773  ;
;2774  ; EXPECTED RESULT HI
;2775  ; RECEIVED RESULT HI
;2776  ; EXPECTED RESULT LO
;2777  ; RECEIVED RESULT LO
;2778  ; ADDRESS OF SP1 HI
;2779  ; TEST FLAG:
;2780  ;  (2) EXPONENT OVERFLOW
;2781  ;  (1) NON_ZERO FRACTION NO OVERFLOW
;2782  ;  (0) ZERO FRACTION NO OVERFLOW
;2783  ; SYNC POINT DESCRIPTION
;2784  ;
;2785  ;-
;2786  ;*****************************************************************
;2787
;2788
```

```
            ;2789 =0
U 1074. 0000.003D.0180.0800.0000.10C8  ;2790 EXPOVF: NEWTST
U 1075. 0818.0038.1180.0800.0000.1215  ;2791  D_K[.4]
U 1215. 0019.0014.0980.09F8.0000.1216  ;2792  ALU_D+K[.2],RC[0F]_ALU
U 1216. 0818.0038.CD80.0800.0000.1217  ;2793  D_K[.F0]
U 1217. 0819.0030.8580.0800.0000.1218  ;2794  D_D.OR.K[.C]
U 1218. 0000.003C.01F8.0800.0084.7219  ;2795  SC_K[.8],Q_0
U 1219. 0D00.003C.0180.0800.0000.121A  ;2796  D_DAL.SC
U 121A. 0001.003C.0180.0A90.0000.121B  ;2797  R[2]_D     ; SAVE EXPECTED RESULT OF NZF TEST
U 121B. 0818.0038.7980.0800.0000.121C  ;2798  D_K[.30]
U 121C. 0001.003C.0180.09D0.0000.121D  ;2799  RC[0A]_D    ; ADDRESS OF SP1 HI
U 121D. 0818.0038.4580.0800.0000.121E  ;2800  D_K[.8000]
U 121E. 0001.003C.0180.09F0.0000.121F  ;2801  RC[0E]_D    ; SAVE EXPECTED RESULT HI
U 121F. 0003.003C.0180.09E0.0000.1220  ;2802  RC[0C]_0    ; SAVE EXPECTED RESULT LO
U 1220. 0018.0038.0980.09C8.0000.1221  ;2803  RC[9]_K[.2]    ; TEST FLAG
U 1221. 0818.0038.4D80.0800.0000.1222  ;2804  D_K[.FF00]
U 1222. 0819.0030.4180.0800.0000.1223  ;2805  D_D.OR.K[.80]
U 1223. 0001.0028.0180.0AA0.0000.1224  ;2806  R[4]_NOT.D    ; MASK FOR BITS <15:07>
U 1224. 0018.0038.7580.0800.0200.1076  ;2807  VA_K[.20]
U 1076. 0000.003D.0180.0800.0000.111F  ;2808 =0 CALL[LOADIB]    ; ADDD R,R
U 1077. 0000.003C.0180.0800.0000.1078  ;2809  NOP
U 1078. 0000.003D.0180.0800.0000.10DA  ;2810 =0 ERLOOP
U 1079. 0000.003C.0180.0800.0000.1225  ;2811  NOP
            ;2812
            ;2813 ; TEST EXPONENT OVERFLOW
            ;2814
U 1225. 0010.0038.0180.0950.0200.1226  ;2815  VA_RC[0A]
U 1226. 0000.003C.0180.4000.0000.1227  ;2816  D[LONG]_CACHE    ; FETCH SP1 HI
U 1227. 0001.003C.0180.0AA8.0000.1228  ;2817  R[5]_D     ; SAVE TO LOAD SA,LA.
U 1228. 0000.003C.01C0.0A20.0000.1229  ;2818  Q_R[4]
U 1229. 001D.0034.0180.0AB0.0000.122A  ;2819  ALU_D.AND.Q,R[6]_ALU   ; MASK OUT SIGN AND EXPONENT
            ;2820       ; SAVE FOR FAD AR1
U 122A. 0003.003C.0180.0AB8.0000.122B  ;2821  R[7]_0     ; SAVE FOR FAD AR0 (SP1 LO)
U 122B. 0000.003C.0180.0803.0000.122C  ;2822  VA_VA+4
U 122C. 0000.003C.0180.4000.0000.122D  ;2823  D[LONG]_CACHE    ; FETCH SP2 HI
U 122D. 0001.003C.0180.0A98.0000.122E  ;2824  R[3]_D     ; SAVE TO LOAD SB,LB
U 122E. 001D.0034.0180.0AC0.0000.1230  ;2825  ALU_D.AND.Q,R[8]_ALU   ; MASK OUT SIGN AND EXPONENT
            ;2826       ; SAVE FOR FAD BR1
U 1230. 0000.003C.01F8.0800.0000.1231  ;2827  Q_0     ; FRACTION BR1=0.111....
U 1231. 0001.2028.0180.0AC8.0000.1080  ;2828  R[9]_NOT.Q,J/EXP1   ; FRACTION BR0=FFFFFFFF
            ;2829
            ;2830 ; CHECK TEST FLAG
            ;2831
U 1232. 0810.0038.0180.0948.0000.1233  ;2832 FLGCHK: D_RC[9]
U 1233. 0019.0000.0580.09C8.0010.1234  ;2833  ALU_D-K[.1],RC[9]_ALU,CLK.UBCC
U 1234. 0000.013C.0180.0800.0000.107A  ;2834  Z?
U 107A. 0000.003C.0180.0800.0000.1235  ;2835 =0 J/NZF
U 107B. 0000.003C.0180.0800.0000.107E  ;2836  J/ZF
            ;2837
            ;2838
            ;2839 ; TEST NO OVERFLOW, NON-ZERO FRACTION
            ;2840 ; SA=0,LA=FF,SB=0,LB=FE
            ;2841 ; AR=0.10......0,BR=0.1111......0
            ;2842
            ;2843
```

```
U 1235. 0000,003C,0180,0803,0000,1236  ;2844 NZF: VA_VA+4
U 1236. 0000,003C,0180,4000,0000,1237  ;2845  D[LONG]_CACHE
U 1237. 0001,003C,0180,0AC8,0000,1238  ;2846  R[9]_D     ; BR0=FFFFFFFE
U 1238. 0000,003C,0180,0803,0000,1239  ;2847  VA_VA+4
U 1239. 0000,003C,0180,4000,0000,123A  ;2848  D[LONG]_CACHE
U 123A. 0001,003C,0180,09F0,0000,107C  ;2849  RC[0E]_D    ; RESULT HI=FFFF7FFF
U 107C. 0000,003D,0180,0800,0000,10DA  ;2850 =0 ERLOOP
U 107D. 0F00,003C,0180,0800,0000,123B  ;2851  D_0
U 123B. 0001,0028,0180,09E0,0000,1080  ;2852  RC[0C]_NOT.D,J/EXP1   ; RESULT LO=FFFFFFFF
           :2853
           :2854
           :2855
           :2856
           :2857 ; TEST NO OVERFLOW ,ZERO FRACTION
           :2858 ; SA=0,LA=FF,SB=0,LB=FF
           :2859 ; AR=0.10....0,BR=0.10...0
           :2860
           :2861
           :2862 =0
U 107E. 0000,003D,0180,0800,0000,10DA  ;2863 ZF: ERLOOP
U 107F. 0003,003C,0180,0AB0,0000,123C  ;2864  R[6]_0     ; AR1=0.1000...
U 123C. 0800,003C,0180,0A28,0000,123D  ;2865  D_R[5]
U 123D. 0019,0030,4580,0AA8,0000,123E  ;2866  ALU_D.OR.K[.8000],R[5]_ALU ; SA=1,SB=0
U 123E. 0818,0038,4180,0800,0000,123F  ;2867  D_K[.80]
U 123F. 0000,003C,01C0,0A18,0000,1240  ;2868  Q_R[3]
U 1240. 001D,0014,0180,0A98,0000,1241  ;2869  ALU_D+Q,R[3]_ALU  ; SET LB=LA
U 1241. 0003,003C,0180,0AC0,0000,1242  ;2870  R[8]_0     ; BR1=0.1000...
U 1242. 0003,003C,0180,0AC8,0000,1243  ;2871  R[9]_0     ; BR0=0000...
U 1243. 0003,003C,0180,09F0,0000,1245  ;2872  RC[0E]_0    ; EXPECTED RESULT HI
U 1245. 0003,003C,0180,09E0,0000,1080  ;2873  RC[0C]_0,J/EXP1    ; EXPECTED RESULT LO
           :2874
           :2875 ; LOAD OPERANDS INTO THE FAD AR,BR LATCHES.
           :2876 ; LOAD SIGNS AND EXPONENTS
           :2877
           :2878 =0
U 1080. 0000,003D,0180,0800,0000,10E4  ;2879 EXP1: CALL[LDFAD]
U 1081. 0000,003C,0180,0800,0000,1082  ;2880  NOP
           :2881
           :2882 ; PERFORM FAD CONDITIONAL ADD/SUB, NORMALIZE AND UPDATE SIGN AND EXPONENT
           :2883
           :2884 =0
U 1082. 0000,003D,0180,0800,0000,101A  ;2885 NRM: CALL[NORM]
U 1083. 0000,003C,0180,0800,0000,1246  ;2886  NOP
           :2887
           :2888
           :2889 ; TEST RESULTS
           :2890
U 1246. 0810,0038,0180,0970,0000,1247  ;2891  D_RC[0E]    ; FETCH EXPECTED RESULT HI
U 1247. 0010,0038,01C0,0968,0000,1248  ;2892  Q_RC[0D]    ; RECEIVED RESULT HI
U 1248. 001D,0000,0180,0800,0010,1249  ;2893  ALU_D-Q,CLK.UBCC   ; CHECK RESULT HI
U 1249. 0000,013C,0180,0800,0000,1084  ;2894  Z?
U 1084. 0000,003D,0180,0800,0000,10DB  ;2895 =0 ERROR1
U 1085. 0810,0038,0180,0960,0000,124A  ;2896  D_RC[0C]    ; FETCH EXPECTED RESULT LO
U 124A. 0010,0038,01C0,0958,0000,124B  ;2897  Q_RC[0B]    ; RECEIVED RESULT LO
U 124B. 001D,0000,0180,0800,0010,124C  ;2898  ALU_D-Q,CLK.UBCC   ; CHECK RESULT LO
```

```
U 124C. 0000.013C.0180.0800.0000.1086  ;2899  Z?
U 1086. 0000.003D.0180.0800.0000.10DB  ;2900 =0 ERROR1
U 1087. 0010.0038.0180.0948.0010.124D  ;2901  ALU_RC[9],CLK.UBCC
U 124D. 0000.013C.0180.0800.0000.1088  ;2902  Z?
U 1088. 0000.003C.0180.0800.0000.1232  ;2903 =0 J/FLGCHK
U 1089. 0000.003C.0180.0800.0000.108A  ;2904  NOP
            ;2905
            ;2906
            ;2907
            ;2908
            ;2909 ; DATA FOR SWR TEST.
            ;2910
            ;2911 ;%10
            ;2912
            ;2913 ;$ 9800. 0000. 0000. 0006
            ;2914 ;$ 1800. 0000. 0000. 0004
            ;2915
            ;2916
            ;2917
            ;2918 ;%20
            ;2919
            ;2920 ; ADDD R,R
            ;2921
            ;2922 ;$ 5460. 0056
            ;2923
            ;2924 ;%30
            ;2925
            ;2926 ;$ 7F80. 0000
            ;2927 ;$ 7F7F. FFFF
            ;2928 ;$ FFFF. FFFE
            ;2929 ;$ 7FFF. FFFF
            ;2930
            ;2931
            ;2932
            ;2933
            ;2934
            ;2935 ; DATA TABLE OF 48 OPERANDS FOR ADDD R,R.
            ;2936 ; NOTE:
            ;2937 ;  (1)  THE DATA TABLE SHOWS THE SP1, AND SP2 OPERANDS
            ;2938 ; WITH HI AND LO WORDS REVERSED.
            ;2939 ;
            ;2940 ; EXAMPLE: TABLE ENTRY SP1  C0800000
            ;2941 ;   ACTUAL DATA SP1  0000C080
            ;2942 ;
            ;2943 ;
            ;2944
            ;2945
            ;2946
            ;2947 ;       SP1           SP2           SP1           SP2
            ;2948 ; C080. 0000.   4040. 0000.   C080. 0000.   4000. 0000
            ;2949 ; C100. 0000.   C000. 0000.   C080. 0000.   C000. 0000
            ;2950 ; C000. 0000.   C000. 0000.   C000. 0000.   C040. 0000
            ;2951 ; C040. 0000.   C040. 0000.   C060. 0000.   C060. 0000
            ;2952
            ;2953
```

```
;2954 ;x40
;2955
;2956
;2957
;2958 ;$ 437C. 0000.   C37E. 0000.   42F8. 0000.   C2FC. 0000
;2959 ;$ 4270. 0000.   C278. 0000.   4270. 0000.   C27C. 0000
;2960 ;$ 41E0. 0000.   C1F0. 0000.   41E8. 0000.   C1FC. 0000
;2961 ;$ 41E0. 0000.   C1F8. 0000.   41E0. 0000.   C1FC. 0000
;2962
;2963 ;80
;2964 ;$ 457F. C000.   C57F. E000.   44FF. 8000.   C4FF. C000
;2965 ;$ 447F. 0000.   C47F. 8000.   447F. 0000.   C47F. C000
;2966 ;$ 43FE. 0000.   C3FF. 0000.   43FE. 8000.   C3FF. C000
;2967 ;$ 43FE. 0000.   C3FF. 8000.   43FE. 0000.   C3FF. C000
;2968
;2969 ;C0
;2970 ;$ 477F. FC00.   C77F. FE00.   46FF. F800.   C6FF. FC00
;2971 ;$ 467F. F000.   C67F. F800.   467F. F000.   C67F. FC00
;2972 ;$ 45FF. E000.   C5FF. F000.   45FF. E800.   C5FF. FC00
;2973 ;$ 45FF. E000.   C5FF. F800.   45FF. E000.   C5FF. FC00
;2974
;2975
;2976 ;100
;2977 ;$ 497F. FFC0.   C97F. FFE0.   48FF. FF80.   C8FF. FFC0
;2978 ;$ 487F. FF00.   C87F. FF80.   487F. FF00.   C87F. FFC0
;2979 ;$ 47FF. FE00.   C7FF. FF00.   47FF. FE80.   C7FF. FFC0
;2980 ;$ 47FF. FE00.   C7FF. FF80.   47FF. FE00.   C7FF. FFC0
;2981
;2982 ;140
;2983 ;$ 4B7F. FFFC.   CB7F. FFFE.   4AFF. FFF8.   CAFF. FFFC
;2984 ;$ 4A7F. FFF0.   CA7F. FFF8.   4A7F. FFF0.   CA7F. FFFC
;2985 ;$ 49FF. FFE0.   C9FF. FFF0.   49FF. FFE8.   C9FF. FFFC
;2986 ;$ 49FF. FFE0.   C9FF. FFF8.   49FF. FFE0.   C9FF. FFFC
;2987
;2988
;2989
;2990 ; RESULTS TABLE FOR THE (48) ADDD R,R OPERATIONS.
;2991 ; NOTE:
;2992 ;  (1)  THE DATA TABLE SHOWS THE RESULT OPERANDS
;2993 ; WITH HI AND LO WORDS REVERSED.
;2994 ;
;2995 ; EXAMPLE: TABLE ENTRY SP1  BF800000
;2996 ;   ACTUAL DATA SP1  0000BF80
;2997 ;
;2998
;2999 ;
;3000
;3001
;3002 ; BF80. 0000. C000. 0000. C120. 0000. C0C0. 0000
;3003 ; C080. 0000. C0A0. 0000. C0C0. 0000. C0E0. 0000
;3004
;3005
;3006 ;x200
;3007
;3008
```

```
;3009 ;$ C000, 0000, C000, 0000, C000, 0000, C040, 0000
;3010 ;$ C000, 0000, C020, 0000, C040, 0000, C060, 0000
;3011
;3012
;3013 ;220
;3014 ;$ C000, 0000, C000, 0000, C000, 0000, C040, 0000
;3015 ;$ C000, 0000, C020, 0000, C040, 0000, C060, 0000
;3016
;3017
;3018 ;240
;3019 ;$ C000, 0000, C000, 0000, C000, 0000, C040, 0000
;3020 ;$ C000, 0000, C020, 0000, C040, 0000, C060, 0000
;3021
;3022 ;260
;3023 ;$ C000, 0000, C000, 0000, C000, 0000, C040, 0000
;3024 ;$ C000, 0000, C020, 0000, C040, 0000, C060, 0000
;3025
;3026
;3027 ;280
;3028 ;$ C000, 0000, C000, 0000, C000, 0000, C040, 0000
;3029 ;$ C000, 0000, C020, 0000, C040, 0000, C060, 0000
;3030
;3031
;3032
;3033 ; DATA TABLE OF 64 OPERANDS FOR ADDD R,R.
;3034 ; NOTE:
;3035 ;  (1)  THE DATA TABLE SHOWS THE SP1 HI,LO, AND
;3036 ; SP2 HI,LO OPERANDS WITH HI AND LO WORDS
;3037 ; REVERSED.
;3038 ;
;3039 ; EXAMPLE: TABLE ENTRY SP1 HI  C0000000
;3040 ;   ACTUAL DATA SP1 HI  0000C000
;3041 ;
;3042 ;
;3043 ;
;3044
;3045 ;
;3046 ;
;3047
;3048
;3049
;3050 ;%300
;3051
;3052
;3053 ;    SP1   HI       SP1   LO       SP2   HI       SP2   LO
;3054 ;$ 4D7F, FFFF,   C000, 0000,   CD7F, FFFF,   E000, 0000
;3055 ;$ 4CFF, FFFF,   8000, 0000,   CCFF, FFFF,   C000, 0000
;3056 ;$ 4C7F, FFFF,   0000, 0000,   CC7F, FFFF,   8000, 0000
;3057 ;$ 4C7F, FFFF,   0000, 0000,   CC7F, FFFF,   C000, 0000
;3058 ;$ 4BFF, FFFE,   0000, 0000,   CBFF, FFFF,   0000, 0000
;3059 ;$ 4BFF, FFFE,   8000, 0000,   CBFF, FFFF,   C000, 0000
;3060 ;$ 4BFF, FFFE,   0000, 0000,   CBFF, FFFF,   8000, 0000
;3061 ;$ 4BFF, FFFE,   0000, 0000,   CBFF, FFFF,   C000, 0000
;3062
;3063
```

```
;3064 ; 380
;3065 ;$ 4F7F. FFFF.   FC00. 0000.   CF7F. FFFF.   FE00. 0000
;3066 ;$ 4EFF. FFFF.   F800. 0000.   CEFF. FFFF.   FC00. 0000
;3067 ;$ 4E7F. FFFF.   F000. 0000.   CE7F. FFFF.   F800. 0000
;3068 ;$ 4E7F. FFFF.   F000. 0000.   CE7F. FFFF.   FC00. 0000
;3069 ;$ 4DFF. FFFF.   E000. 0000.   CDFF. FFFF.   F000. 0000
;3070 ;$ 4DFF. FFFF.   E800. 0000.   CDFF. FFFF.   FC00. 0000
;3071 ;$ 4DFF. FFFF.   E000. 0000.   CDFF. FFFF.   F800. 0000
;3072 ;$ 4DFF. FFFF.   E000. 0000.   CDFF. FFFF.   FC00. 0000
;3073
;3074
;3075 ; 400
;3076 ;$ 517F. FFFF.   FFC0. 0000.   D17F. FFFF.   FFE0. 0000
;3077 ;$ 50FF. FFFF.   FF80. 0000.   D0FF. FFFF.   FFC0. 0000
;3078 ;$ 507F. FFFF.   FF00. 0000.   D07F. FFFF.   FF80. 0000
;3079 ;$ 507F. FFFF.   FF00. 0000.   D07F. FFFF.   FFC0. 0000
;3080 ;$ 4FFF. FFFF.   FE00. 0000.   CFFF. FFFF.   FF00. 0000
;3081 ;$ 4FFF. FFFF.   FE80. 0000.   CFFF. FFFF.   FFC0. 0000
;3082 ;$ 4FFF. FFFF.   FE00. 0000.   CFFF. FFFF.   FF80. 0000
;3083 ;$ 4FFF. FFFF.   FE00. 0000.   CFFF. FFFF.   FFC0. 0000
;3084
;3085
;3086 ; 480
;3087 ;$ 537F. FFFF.   FFFC. 0000.   D37F. FFFF.   FFFE. 0000
;3088 ;$ 52FF. FFFF.   FFF8. 0000.   D2FF. FFFF.   FFFC. 0000
;3089 ;$ 527F. FFFF.   FFF0. 0000.   D27F. FFFF.   FFF8. 0000
;3090 ;$ 527F. FFFF.   FFF0. 0000.   D27F. FFFF.   FFFC. 0000
;3091 ;$ 51FF. FFFF.   FFE0. 0000.   D1FF. FFFF.   FFF0. 0000
;3092 ;$ 51FF. FFFF.   FFE8. 0000.   D1FF. FFFF.   FFFC. 0000
;3093 ;$ 51FF. FFFF.   FFE0. 0000.   D1FF. FFFF.   FFF8. 0000
;3094 ;$ 51FF. FFFF.   FFE0. 0000.   D1FF. FFFF.   FFFC. 0000
;3095
;3096
;3097 ; 500
;3098 ;$ 557F. FFFF.   FFFF. C000.   D57F. FFFF.   FFFF. E000
;3099 ;$ 54FF. FFFF.   FFFF. 8000.   D4FF. FFFF.   FFFF. C000
;3100 ;$ 547F. FFFF.   FFFF. 0000.   D47F. FFFF.   FFFF. 8000
;3101 ;$ 547F. FFFF.   FFFF. 0000.   D47F. FFFF.   FFFF. C000
;3102 ;$ 53FF. FFFF.   FFFE. 0000.   D3FF. FFFF.   FFFF. 0000
;3103 ;$ 53FF. FFFF.   FFFE. 8000.   D3FF. FFFF.   FFFF. C000
;3104 ;$ 53FF. FFFF.   FFFE. 0000.   D3FF. FFFF.   FFFF. 8000
;3105 ;$ 53FF. FFFF.   FFFE. 0000.   D3FF. FFFF.   FFFF. C000
;3106
;3107
;3108 ; 580
;3109 ;$ 577F. FFFF.   FFFF. FC00.   D77F. FFFF.   FFFF. FE00
;3110 ;$ 56FF. FFFF.   FFFF. F800.   D6FF. FFFF.   FFFF. FC00
;3111 ;$ 567F. FFFF.   FFFF. F000.   D67F. FFFF.   FFFF. F800
;3112 ;$ 567F. FFFF.   FFFF. F000.   D67F. FFFF.   FFFF. FC00
;3113 ;$ 55FF. FFFF.   FFFF. E000.   D5FF. FFFF.   FFFF. F000
;3114 ;$ 55FF. FFFF.   FFFF. E800.   D5FF. FFFF.   FFFF. FC00
;3115 ;$ 55FF. FFFF.   FFFF. E000.   D5FF. FFFF.   FFFF. F800
;3116 ;$ 55FF. FFFF.   FFFF. E000.   D5FF. FFFF.   FFFF. FC00
;3117
;3118 ; 600
```

```
;3119 ;$ 597F. FFFF.   FFFF. FFC0.   D97F. FFFF.   FFFF. FFE0
;3120 ;$ 58FF. FFFF.   FFFF. FF80.   D8FF. FFFF.   FFFF. FFC0
;3121 ;$ 587F. FFFF.   FFFF. FF00.   D87F. FFFF.   FFFF. FF80
;3122 ;$ 587F. FFFF.   FFFF. FF00.   D87F. FFFF.   FFFF. FFC0
;3123 ;$ 57FF. FFFF.   FFFF. FE00.   D7FF. FFFF.   FFFF. FF00
;3124 ;$ 57FF. FFFF.   FFFF. FE80.   D7FF. FFFF.   FFFF. FFC0
;3125 ;$ 57FF. FFFF.   FFFF. FE00.   D7FF. FFFF.   FFFF. FF80
;3126 ;$ 57FF. FFFF.   FFFF. FE00.   D7FF. FFFF.   FFFF. FFC0
;3127
;3128 ; 680
;3129 ;$ 5B7F. FFFF.   FFFF. FFFC.   DB7F. FFFF.   FFFF. FFFE
;3130 ;$ 5AFF. FFFF.   FFFF. FFF8.   DAFF. FFFF.   FFFF. FFFC
;3131 ;$ 5A7F. FFFF.   FFFF. FFF0.   DA7F. FFFF.   FFFF. FFF8
;3132 ;$ 5A7F. FFFF.   FFFF. FFF0.   DA7F. FFFF.   FFFF. FFFC
;3133 ;$ 59FF. FFFF.   FFFF. FFE0.   D9FF. FFFF.   FFFF. FFF0
;3134 ;$ 59FF. FFFF.   FFFF. FFE8.   D9FF. FFFF.   FFFF. FFFC
;3135 ;$ 59FF. FFFF.   FFFF. FFE0.   D9FF. FFFF.   FFFF. FFF8
;3136 ;$ 59FF. FFFF.   FFFF. FFE0.   D9FF. FFFF.   FFFF. FFFC
;3137
;3138
;3139 ; RESULTS TABLE FOR THE (64) ADDD R,R OPERATIONS.
;3140 ; NOTE:
;3141 ;  (1) THE RESULTS ARE QUAD WORDS FOR WHICH THE LEAST
;3142 ; SIGNIFICANT WORD IS ZERO. THE LO HALF OF THE MSW
;3143 ; IS ALSO ZERO, AND IS NOT SHOWN IN THE TABLE.
;3144 ;
;3145 ;  (2)  THE DATA TABLE SHOWS THE RESULT OPERANDS
;3146 ; WITH HI AND LO WORDS REVERSED.
;3147 ;
;3148 ; EXAMPLE: TABLE ENTRY SP1  C0400000
;3149 ;    ACTUAL DATA SP1  0000C040
;3150 ;
;3151
;3152 ;
;3153 ;x800
;3154
;3155
;3156 ;$ C000. 0000. C000. 0000. C000. 0000. C040. 0000
;3157 ;$ C000. 0000. C020. 0000. C040. 0000. C060. 0000
;3158
;3159
;3160
;3161
```

```
                ;3162  .PAGE "TEST 28B ACCELERATOR OVERRIDE TEST (1)"
                ;3163  ;*****************************************************************
                ;3164  ;**
                ;3165  ; ACCELERATOR OVERRIDE TEST (1)
                ;3166  ;
                ;3167  ; TEST DESCRIPTION
                ;3168  ;
                ;3169  ; THE IB IS LOADED WITH THE APPRPRIATE INSTRUCTION, THAT
                ;3170  ; GENERATES FPA OVERRIDE AT THE DESIRED EXECUTION POINT
                ;3171  ; COUNT.
                ;3172  ;
                ;3173  ; FPA UCODE USED
                ;3174  ;
                ;3175  ; LOCATION CONTENTS
                ;3176  ; 0A3: SCR/R.R,FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
                ;3177  ;
                ;3178  ; LOGIC DESCRIPTION
                ;3179  ;
                ;3180  ; THE ACELERATOR OVERRIDE LOGIC IS IN THE FCT MODULE.
                ;3181  ;
                ;3182  ;
                ;3183  ;
                ;3184  ; ERROR DESCRIPTION
                ;3185  ;
                ;3186  ; EXPECTED CPU BRANCH ADDRESS
                ;3187  ; RECEIVED CPU BRANCH ADDRESS
                ;3188  ; ADDRESS OF FIRST IB LOAD
                ;3189  ; SYNC POINT DESCRIPTION
                ;3190  ;
                ;3191  ;--
                ;3192  ;*****************************************************************
                ;3193  ;/////////////////////////////////////////////////////////////////
                ;3194  ;
                ;3195  ; NOTE*****
                ;3196  ;
                ;3197  ; THIS TEST ISSUES ACCELERATOR OVERRIDE TO THE CPU
                ;3198  ; IF THIS SIGNAL IS NOT ASSERTED (DUE TO SOME FAILURE)
                ;3199  ; THE CPU USEQUENCER WILL JUMP OUT OF WCS INTO PCS. THE
                ;3200  ; TEST WILL TIME OUT. ENTER THE CONTINUE COMMAND TO
                ;3201  ; GO TO THE NEXT TEST.
                ;3202  ;
                ;3203  ;
                ;3204  ;/////////////////////////////////////////////////////////////////
                ;3205
                ;3206
                ;3207
                ;3208  =0
U 108A, 0018.0039,0D80,09F8,0000,10C8  ;3209 ACCOVR: NEWTST[.3]
U 108B, 0818.0038,D980,0800,0000,124E  ;3210  D_K[.9]
U 124E, 000B.003C,01F8,0800,0084,724F  ;3211  SC_K[.8],Q_0
U 124F, 0D0B.003C,0180,0800,0000,1250  ;3212  D_DAL.SC
U 1250, 0001.003C,0180,09E0,0000,1251  ;3213  RC[0C]_D    ; ADDRESS OF FIRST IBLOAD (900)
U 1251, 0818.0038,E980,0800,0000,1252  ;3214  D_K[.24]
U 1252, 000B.003C,5DF8,0800,0084,7253  ;3215  SC_K[.7],Q_0
U 1253, 0D0B.003C,0180,0800,0000,1254  ;3216  D_DAL.SC
```

```
U 1254. 0001.003C.0180.0AE0.0000.108C  ;3217  R[0C] D    ; CONSTANT 1200.
U 108C. 0000.003D.0180.0800.0000.10DA  ;3218 =0 ERLOOP
U 108D. 0000.003C.0180.0800.0000.108E  ;3219  NOP
          ;3220
          ;3221 ; TEST FPA OVERRIDE  AT THE BEGINNING OF A FORK.
          ;3222 ; TWO ADDRESS INSTRUCTION, ADDF R,R.
          ;3223
U 108E. 0000.003D.0180.0800.0000.1114  ;3224 =0 CALL[FPINIT]   ; INIT FPA
U 108F. 0000.003C.0180.0800.0000.1090  ;3225  NOP
U 1090. 0000.003D.0180.0800.0000.1024  ;3226 =0 CALL[FPIRD]
U 1091. 0010.0038.0180.0960.0200.1092  ;3227  VA_RC[0C]
U 1092. 0000.003D.0180.0800.0000.111F  ;3228 =0 CALL[LOADIB]   ; OPCODE ADDF R,R
U 1093. 0000.003C.0180.0800.0000.1004  ;3229  NOP
U 1004. 0000.003D.0180.0800.0000.1005  ;3230 =00 CALL
U 1005. 0000.003F.0180.0800.0000.0200  ;3231  J/0200,SUB/SPEC
          ;3232
          ;3233 ; FPA ISSUES ACCELERATOR OVERRIDE
          ;3234 ; CHECK CPU RESPONCE TO FPA OVERRIDE.
          ;3235
U 1006. 0000.003C.01C0.0A60.0000.1007  ;3236  Q_R[0C]
U 1007. 0019.2030.05C0.0800.0000.1255  ;3237  ALU_Q.OR.K[.1],Q_ALU
U 1255. 0001.203C.0180.09F0.0000.1256  ;3238  RC[0E] Q  ; SAVE EXPECTED CPU BRANCH ADDRESS
U 1256. 0810.0038.0180.0968.0000.1257  ;3239  D_RC[0D]  ; RECEIVED CPU BRANCH ADDRESS
U 1257. 001D.2000.0180.0800.0010.1258  ;3240  ALU_Q-D,CLK.UBCC
U 1258. 0000.013C.0180.0800.0000.1094  ;3241  Z?
U 1094. 0000.003D.0180.0800.0000.10DB  ;3242 =0 ERROR1
U 1095. 0000.003C.0180.0800.0000.1096  ;3243  NOP
          ;3244
```

```
                ;3245  .PAGE "TEST 28C ACCELERATOR OVERRIDE TEST (2)"
                ;3246  ;*****************************************************************
                ;3247  ;**
                ;3248  ; ACCELERATOR OVERRIDE TEST (2)
                ;3249  ;
                ;3250  ; TEST DESCRIPTION
                ;3251  ;
                ;3252  ; THE IB IS LOADED WITH THE APPRPRIATE INSTRUCTION, THAT
                ;3253  ; GENERATES FPA OVERRIDE AT THE DESIRED EXECUTION POINT
                ;3254  ; COUNT.
                ;3255  ;
                ;3256  ; FPA UCODE USED
                ;3257  ;
                ;3258  ; LOCATION CONTENTS
                ;3259  ; 0A3: SCR/R.R,FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
                ;3260  ;
                ;3261  ; LOGIC DESCRIPTION
                ;3262  ;
                ;3263  ; THE ACELERATOR OVERRIDE LOGIC IS IN THE FCT MODULE.
                ;3264  ;
                ;3265  ;
                ;3266  ;
                ;3267  ; ERROR DESCRIPTION
                ;3268  ;
                ;3269  ; EXPECTED CPU BRANCH ADDRESS
                ;3270  ; RECEIVED CPU BRANCH ADDRESS
                ;3271  ; ADDRESS OF FIRST IB LOAD
                ;3272  ; SYNC POINT DESCRIPTION
                ;3273  ;
                ;3274  ;--
                ;3275  ;*****************************************************************************
                ;3276  ;/////////////////////////////////////////////////////////////////////////////
                ;3277  ;
                ;3278  ; NOTE*****
                ;3279  ;
                ;3280  ; THIS TEST ISSUES ACCELERATOR OVERRIDE TO THE CPU
                ;3281  ; IF THIS SIGNAL IS NOT ASSERTED (DUE TO SOME FAILURE)
                ;3282  ; THE CPU USEQUENCER WILL JUMP OUT OF WCS INTO PCS. THE
                ;3283  ; TEST WILL TIME OUT. ENTER THE CONTINUE COMMAND TO
                ;3284  ; GO TO THE NEXT TEST.
                ;3285  ;
                ;3286  ;
                ;3287  ;/////////////////////////////////////////////////////////////////////////////
                ;3288
                ;3289
                ;3290
                ;3291  =0
U 1096. 0018.0039.0D80.09F8.0000.10C8  ;3292 ACCOVR1: NEWTST[.3]
U 1097. 0818.0038.D980.0800.0000.1259  ;3293  D_K[.9]
U 1259. 0000.003C.01F8.0800.0084.725A  ;3294  SC_K[.8].Q_0
U 125A. 0D00.003C.0180.0800.0000.125B  ;3295  D_DAL.SC
U 125B. 0019.0030.1180.09E0.0000.125C  ;3296  ALU_D.OR.K[.4],RC[0C]_ALU ; ADDRESS OF IBLOAD (904)
U 125C. 0818.0038.E980.0800.0000.125D  ;3297  D_K[.24]
U 125D. 0000.003C.5DF8.0800.0084.725E  ;3298  SC_K[.7].Q_0
U 125E. 0D00.003C.0180.0800.0000.125F  ;3299  D_DAL.SC
```

```
U 125F. 0001.003C.0180.0AE0.0000.109B  ;3300  R[0C]_D    ; CONSTANT 1200.
U 1098. 0000.003D.0180.0800.0000.10DA  ;3301 =0 ERLOOP
U 1099. 0000.003C.0180.0800.0000.109A  ;3302  NOP
          ;3303
          ;3304 ; TEST FOR FPA OVERRIDE AT THE BEGINNING OF B FORK..
          ;3305 ; TWO ADDRESS INSTRUCTION, ADDF M,R
          ;3306
U 109A. 0000.003D.0180.0800.0000.1114  ;3307 =0 CALL[FPINIT]   ; INIT FPA
U 109B. 0000.003C.0180.0800.0000.109C  ;3308  NOP
U 109C. 0000.003D.0180.0800.0000.1024  ;3309 =0 CALL[FPIRD]
U 109D. 0810.0038.0180.0960.0000.1260  ;3310  D_RC[0C]
U 1260. 0010.0038.0180.0960.0200.109E  ;3311  VA_RC[0C]
U 109E. 0000.003D.0180.0800.0000.111F  ;3312 =0 CALL[LOADIB]   ; OPCODE ADDF M,R
U 109F. D000.003C.0180.0800.0000.1008  ;3313  IBC/DD    ; ADVANCE EP COUNTER TO (1)
          ;3314
U 1008. 0000.003D.0180.0800.0000.1009  ;3315 =00 CALL
U 1009. 0000.003F.0180.0800.0000.0200  ;3316  J/0200,SUB/SPEC
          ;3317
          ;3318 ; FPA ISSUES ACCELERATOR OVERRIDE
          ;3319 ; CHECK CPU RESPONCE TO FPA OVERRIDE.
          ;3320
U 100A. 0000.003C.01C0.0A60.0000.100B  ;3321  Q_R[0C]
U 100B. 0019.2030.09C0.0800.0000.1261  ;3322  ALU_Q.OR.K[.2],Q_ALU
U 1261. 0001.203C.0180.09F0.0000.1262  ;3323  RC[0E]_Q  ; SAVE EXPECTED CPU BRANCH ADDRESS
U 1262. 0810.0038.0180.0968.0000.1263  ;3324  D_RC[0D]  ; RECEIVED CPU BRANCH ADDRESS
U 1263. 001D.2000.0180.0800.0010.1264  ;3325  ALU_Q-D,CLK.UBCC
U 1264. 0000.013C.0180.0800.0000.10A0  ;3326  Z?
U 10A0. 0000.003D.0180.0800.0000.10DB  ;3327 =0 ERROR1
U 10A1. 0000.003C.0180.0800.0000.10A2  ;3328  NOP
          ;3329
```

```
          ;3330  .PAGE "TEST 28D ACCELERATOR OVERRIDE TEST (3)"
          ;3331 ;*****************************************************************
          ;3332 ;**
          ;3333 ; ACCELERATOR OVERRIDE TEST (3)
          ;3334 ;
          ;3335 ; TEST DESCRIPTION
          ;3336 ;
          ;3337 ; THE IB IS LOADED WITH THE APPRPRIATE INSTRUCTION, THAT
          ;3338 ; GENERATES FPA OVERRIDE AT THE DESIRED EXECUTION POINT
          ;3339 ; COUNT.
          ;3340 ;
          ;3341 ; FPA UCODE USED
          ;3342 ;
          ;3343 ; LOCATION CONTENTS
          ;3344 ; 0A3: SCR/R.R,FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
          ;3345 ;
          ;3346 ; LOGIC DESCRIPTION
          ;3347 ;
          ;3348 ; THE ACELERATOR OVERRIDE LOGIC IS IN THE FCT MODULE.
          ;3349 ;
          ;3350 ;
          ;3351 ;
          ;3352 ; ERROR DESCRIPTION
          ;3353 ;
          ;3354 ; EXPECTED CPU BRANCH ADDRESS
          ;3355 ; RECEIVED CPU BRANCH ADDRESS
          ;3356 ; ADDRESS OF FIRST IB LOAD
          ;3357 ; SYNC POINT DESCRIPTION
          ;3358 ;
          ;3359 ;--
          ;3360 ;*****************************************************************
          ;3361 ;/////////////////////////////////////////////////////////////////
          ;3362 ;
          ;3363 ; NOTE*****
          ;3364 ;
          ;3365 ; THIS TEST ISSUES ACCELERATOR OVERRIDE TO THE CPU
          ;3366 ; IF THIS SIGNAL IS NOT ASSERTED (DUE TO SOME FAILURE)
          ;3367 ; THE CPU USEQUENCER WILL JUMP OUT OF WCS INTO PCS. THE
          ;3368 ; TEST WILL TIME OUT. ENTER THE CONTINUE COMMAND TO
          ;3369 ; GO TO THE NEXT TEST.
          ;3370 ;
          ;3371 ;
          ;3372 ;/////////////////////////////////////////////////////////////////
          ;3373
          ;3374
          ;3375
          ;3376 =0
U 10A2, 0018.0039.0D80.09F8.0000.10C8  ;3377 ACCOVR2: NEWTST[.3]
U 10A3, 0818.0038.D980.0800.0000.1265  ;3378  D_K[.9]
U 1265, 0000.003C.01F8.0800.0084.7266  ;3379  SC_K[.8],Q_0
U 1266, 0D00.003C.0180.0800.0000.1267  ;3380  D_DAL.SC
U 1267, 0019.0030.0180.09E0.0000.1268  ;3381  ALU_D.OR.K[.8],RC[0C]_ALU ; ADDRESS OF  IBLOAD (908)
U 1268, 0818.0038.E980.0800.0000.1269  ;3382  D_K[.24]
U 1269, 0000.003C.5DF8.0800.0084.726A  ;3383  SC_K[.7],Q_0
U 126A, 0D00.003C.0180.0800.0000.126B  ;3384  D_DAL.SC
```

```
U 126B, 0001.003C.0180.0AE0.0000.10A4  ;3385  R[0C]_D    ; CONSTANT 1200.
U 10A4, 0000.003D.0180.0800.0000.10DA  ;3386 =0 ERLOOP
U 10A5, 0000.003C.0180.0800.0000.10A6  ;3387  NOP
          ;3388
          ;3389
          ;3390 ; TEST FOR FPA OVERRIDE AT THE BEGINNING OF B FORK..
          ;3391 ; THREE ADDRESS INSTRUCTION ADDF R,R
          ;3392
          ;3393
U 10A6, 0000.003D.0180.0800.0000.1114  ;3394 =0 CALL[FPINIT]   ; INIT FPA
U 10A7, 0000.003C.0180.0800.0000.10A8  ;3395  NOP
U 10A8, 0000.003D.0180.0800.0000.1024  ;3396 =0 CALL[FPIRD]
U 10A9, 0010.0038.0180.0960.0200.10AA  ;3397  VA_RC[0C]
U 10AA, 0000.003D.0180.0800.0000.111F  ;3398 =0 CALL[LOADIB]   ; OPCODE ADDF R,R
U 10AB, D000.003C.0180.0800.0000.100C  ;3399  IBC/0D    ; ADVANCE EP COUNTER TO (1)
U 100C, 0000.003D.0180.0800.0000.100D  ;3400 =00 CALL
U 100D, 0000.003F.0180.0800.0000.0200  ;3401  J/0200,SUB/SPEC
          ;3402
          ;3403 ; FPA ISSUES ACCELERATOR QVERRIDE
          ;3404 ; CHECK CPU RESPONCE TO FPA OVERRIDE.
          ;3405
U 100E, 0000.003C.01C0.0A60.0000.100F  ;3406  Q_R[0C]
U 100F, 0019.2030.05C0.0800.0000.126C  ;3407  ALU_Q.OR.K[.1],Q_ALU
U 126C, 0001.203C.0180.09F0.0000.126D  ;3408  RC[0E]_Q   ; SAVE EXPECTED CPU BRANCH ADDRESS
U 126D, 0810.0038.0180.0968.0000.126E  ;3409  D_RC[0D]   ; RECEIVED CPU BRANCH ADDRESS
U 126E, 001D.2000.0180.0800.0010.126F  ;3410  ALU_Q-D,CLK.UBCC
U 126F, 0000.013C.0180.0800.0000.10AC  ;3411  Z?
U 10AC, 0000.003D.0180.0800.0000.10DB  ;3412 =0 ERROR1
U 10AD, 0000.003C.0180.0800.0000.10AE  ;3413  NOP
          ;3414
```

```
          ;3415  .PAGE "TEST 28E ACCELERATOR OVERRIDE TEST (4)"
          ;3416 ;*****************************************************************
          ;3417 ;++
          ;3418 ; ACCELERATOR OVERRIDE TEST (4)
          ;3419 ;
          ;3420 ; TEST DESCRIPTION
          ;3421 ;
          ;3422 ; THE IB IS LOADED WITH THE APPRPRIATE INSTRUCTION, THAT
          ;3423 ; GENERATES FPA OVERRIDE AT THE DESIRED EXECUTION POINT
          ;3424 ; COUNT.
          ;3425 ;
          ;3426 ; FPA UCODE USED
          ;3427 ;
          ;3428 ; LOCATION CONTENTS
          ;3429 ; 0A3: SCR/R.R,FADC/R1,MSC/IR0,BSC/R,SGNC/LDS,OPLD/LDR.R,EAC/LDAB
          ;3430 ;
          ;3431 ; LOGIC DESCRIPTION
          ;3432 ;
          ;3433 ; THE ACELERATOR OVERRIDE LOGIC IS IN THE FCT MODULE.
          ;3434 ;
          ;3435 ;
          ;3436 ;
          ;3437 ; ERROR DESCRIPTION
          ;3438 ;
          ;3439 ; EXPECTED CPU BRANCH ADDRESS
          ;3440 ; RECEIVED CPU BRANCH ADDRESS
          ;3441 ; ADDRESS OF FIRST IB LOAD
          ;3442 ; SYNC POINT DESCRIPTION
          ;3443 ;
          ;3444 ;--
          ;3445 ;*****************************************************************
          ;3446 ;/////////////////////////////////////////////////////////////////
          ;3447 ;
          ;3448 ; NOTE*****
          ;3449 ;
          ;3450 ; THIS TEST ISSUES ACCELERATOR OVERRIDE TO THE CPU
          ;3451 ; IF THIS SIGNAL IS NOT ASSERTED (DUE TO SOME FAILURE)
          ;3452 ; THE CPU USEQUENCER WILL JUMP OUT OF WCS INTO PCS. THE
          ;3453 ; TEST WILL TIME OUT. ENTER THE CONTINUE COMMAND TO
          ;3454 ; GO TO THE NEXT TEST.
          ;3455 ;
          ;3456 ;
          ;3457 ;/////////////////////////////////////////////////////////////////
          ;3458
          ;3459
          ;3460
          ;3461 =0
U 10AE, 0018.0039.0D80.09F8.0000.10C8  ;3462 ACCOVR3: NEWTST[.3]
U 10AF, 0818.0038.D980.0800.0000.1270  ;3463  D_K[.9]
U 1270, 0000.003C.01F8.0800.0084.7271  ;3464  SC_K[.8],Q_0
U 1271, 0D00.003C.0180.0800.0000.1272  ;3465  D_DAL.SC
U 1272, 0019.0030.8580.09E0.0000.1273  ;3466  ALU_D.OR.K[.C],RC[0C]_ALU ; ADDRESS OF FIRST IBLOAD (90C)
U 1273, 0818.0038.E980.0800.0000.1274  ;3467  D_K[.24]
U 1274, 0000.003C.5DF8.0800.0084.7275  ;3468  SC_K[.7],Q_0
U 1275, 0D00.003C.0180.0800.0000.1276  ;3469  D_DAL.SC
```

```
U 1276, 0001,003C,0180,0AE0,0000,10B0  ;3470  R[0C]_D    ; CONSTANT 1200.
U 10B0, 0000,003D,0180,0800,0000,10DA  ;3471 =0 ERLOOP
U 10B1, 0000,003C,0180,0800,0000,10B2  ;3472  NOP
          ;3473
          ;3474 ; TEST FOR FPA OVERRIDE AT THE BEGINNING OF C FORK
          ;3475 ; EMODF R,R
          ;3476
          ;3477
U 10B2, 0000,003D,0180,0800,0000,1114  ;3478 =0 CALL[FPINIT]   ; INIT FPA
U 10B3, 0000,003C,0180,0800,0000,10B4  ;3479  NOP
U 10B4, 0000,003D,0180,0800,0000,1024  ;3480 =0 CALL[FPIRD]
U 10B5, 0010,0038,0180,0960,0200,10B6  ;3481  VA_RC[0C]
U 10B6, 0000,003D,0180,0800,0000,111F  ;3482 =0 CALL[LOADIB]   ; OPCODE EMODF R,R
U 10B7, D000,003C,0180,0800,0000,1277  ;3483  IBC/0D    ; ADVANCE EP COUNTER TO (1)
U 1277, D000,003C,0180,0800,0000,1010  ;3484  IBC/0D    ; ADVANCE EP COUNTER TO (2)
          ;3485
U 1010, 0000,003D,0180,0800,0000,1011  ;3486 =00 CALL
U 1011, 0000,003F,0180,0800,0000,0200  ;3487  J/0200,SUB/SPEC
          ;3488
          ;3489 ; FPA ISSUES ACCELERATOR QVERRIDE
          ;3490 ; CHECK CPU RESPONCE TO FPA OVERRIDE.
          ;3491
U 1012, 0000,003C,01C0,0A60,0000,1013  ;3492  Q_R[0C]
U 1013, 0019,2030,11C0,0800,0000,1278  ;3493  ALU_Q.OR.K[.4],Q_ALU
U 1278, 0001,203C,0180,09F0,0000,1279  ;3494  RC[0E]_Q   ; SAVE EXPECTED CPU BRANCH ADDRESS
U 1279, 0810,0038,0180,0968,0000,127A  ;3495  D_RC[0D]   ; RECEIVED CPU BRANCH ADDRESS
U 127A, 001D,2000,0180,0800,0010,127B  ;3496  ALU_Q-D,CLK.UBCC
U 127B, 0000,013C,0180,0800,0000,10B8  ;3497  Z?
U 10B8, 0000,003D,0180,0800,0000,10DB  ;3498 =0 ERROR1
U 10B9, 0000,003C,0180,0800,0000,10BA  ;3499  NOP
          ;3500
```

```
ZZ-ESKAR-V22  TEST 28F RESERVED                                                    SEQ 2734
ESKAR63.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:26              Page    8

            ;3501  .PAGE "TEST 28F RESERVED"
            ;3502 =0
U 10BA, 0000,003D,0180,0800,0000,10C8  ;3503 TSTXX3: NEWTST
U 10BB, 0000,003C,0180,0800,0000,10BC  ;3504  NOP
            ;3505
```

```
            ;3506  .PAGE "TEST 290 RESERVED"
            ;3507 =0
U 10BC, 0000,003D,0180,0800,0000,10C8  ;3508 T290: NEWTST
U 10BD, 0000,003C,0180,0800,0000,127C  ;3509  NOP
            ;3510
```

```
            ;3511  .PAGE "DUMMY NEWTST"
U 127C, 0000.003D.0180.0800.0000.10C8  ;3512 TSTXX4: NEWTST
            ;3513
            ;3514
            ;3515
            ;3516 ; DISPATCH TABLE
            ;3517
U 1244, 0800.003C.0180.0A60.0000.127D  ;3518 1244: D_R[0C]
U 127D, 0019.0016.0580.09E8.0000.0002  ;3519  ALU_D+K[.1],RC[0D]_ALU,RETURN2
U 122F, 0800.003C.0180.0A60.0000.127E  ;3520 122F: D_R[0C]
U 127E, 0019.0016.0980.09E8.0000.0002  ;3521  ALU_D+K[.2],RC[0D]_ALU,RETURN2
U 12C2, 0800.003C.0180.0A60.0000.127F  ;3522 12C2: D_R[0C]
U 127F, 0019.0016.1180.09E8.0000.0002  ;3523  ALU_D+K[.4],RC[0D]_ALU,RETURN2
            ;3524
            ;3525
            ;3526
            ;3527
            ;3528
            ;3529 ;%900
            ;3530
            ;3531 ; ADDF2 R,R
            ;3532 ;$ 5640, 0055
            ;3533 ; ADDF2 M,R
            ;3534 ;$ 0040, A556
            ;3535 ; ADDF3 R,R
            ;3536 ;$ 0041, 5556
            ;3537 ; EMODF R,R
            ;3538 ;$ 0055, 0000, 0054, 5600
            ;3539
            ;3540
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2737
ESKAR63.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:26          Page   9
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000   1907: 1874 1881= 1897= 3230= 3231= 3236= 3237=
U 1008   3315= 3316= 3321= 3322= 3400= 3401= 3406= 3407=
U 1010   3486= 3487= 3492= 3493= 1898= 1899= 1918= 1919=
U 1018   1992= 1993= 2021= 2022= 2071= 2072= 2074= 2075=
U 1020   2076= 2077= 2079= 2080= 2099= 2100= 2122= 2123=
U 1028   2124= 2125= 2197= 2198= 2224= 2225= 2226= 2227=
U 1030   2228= 2229= 2235= 2236= 2240= 2241= 2281= 2282=
U 1038   2346= 2347= 2364= 2365= 2367= 2368= 2370= 2371=
U 1040   2393= 2394= 2410= 2411= 2415= 2416= 2426= 2427=
U 1048   2431= 2435= 2438= 2444= 2446= 2447= 1939: 1875
U 1050   2453= 2454= 2460= 2461= 2559= 2560= 2581= 2582=
U 1058   2584= 2585= 1942: 1876 2587= 2588= 1945: 1877
U 1060   2619= 2620= 2633= 2634= 2652= 2653= 2657= 2658=
U 1068   2668= 2669= 2673= 2677= 2680= 2686= 2688= 2689=
U 1070   2695= 2696= 2702= 2703= 2790= 2791= 2808= 2809=
U 1078   2810= 2811= 2835= 2836= 2850= 2851= 2863= 2864=
U 1080   2879= 2880= 2885= 2886= 2895= 2896= 2900= 2901=
U 1088   2903= 2904= 3209= 3210= 3218= 3219= 3224= 3225=
U 1090   3226= 3227= 3228= 3229= 3242= 3243= 3292= 3293=
U 1098   3301= 3302= 3307= 3308= 3309= 3310= 3312= 3313=
U 10A0   3327= 3328= 3377= 3378= 3386= 3387= 3394= 3395=
U 10A8   3396= 3397= 3398= 3399= 3412= 3413= 3462= 3463=
U 10B0   3471= 3472= 3478= 3479= 3480= 3481= 3482= 3483=
U 10B8   3498= 3499= 3503= 3504= 3508= 3509= 1878 1879
U 10C0   1880 1900 1901 1902 1903 1904 1905 1906
U 10C8   1916 1917 1920 1921 1922 1923 1924 1925
U 10D0   1926 1927 1928 1929 1930 1931 1932 1933
U 10D8   1934 1935 1936 1937 1938 1940 1941 1973
U 10E0   1974 1980 1981 1982 1990 1991 1994 1995
U 10E8   1996 1997 1998 1999 2000 2001 2002 2003
U 10F0   2004 2005 2006 2007 2023 2024 2028 2030
U 10F8   2032 2034 2036 2038 2039 2043 2046 2049
U 1100   1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108   1869: 2051 2052 2053 1870: 1871: 1872: 1873:
U 1110   2054 2055 2056 2057 2069 2070 2073 2078
U 1118   2081 2091 2092 2093 2101 2102 2103 2109
U 1120   2110 2111 2112 2113 2114 2115 2116 2117
U 1128   2118 2126 2127 2128 2129 2130 2131 2132
U 1130   2133 2134 2135 2199 2200 2201 2202 2203
U 1138   2204 2205 2206 2207 2208 2209 2210 2211
U 1140   2212 2213 2214 2215 2216 2217 2218 2219
U 1148   2220 2221 2222 2223 2242 2243 2247 2249
U 1150   2251 2253 2255 2258 2262 1965: 2268 2271
U 1158   2276 2277 2278 2279 2280 2348 2349 2350
U 1160   2351 2352 2353 2354 2355 2356 2357 2358
U 1168   2359 2360 2361 2362 2363 2369 2372 2373
U 1170   2374 2375 2377 2378 2379 2380 2381 2383
U 1178   2384 2385 2386 2387 2391 2392 2395 2396
U 1180   2397 2398 2399 2400 2401 2402 2422 2423
U 1188   2424 2425 2428 2429 2430 2436 2437 2445
U 1190   2451 2452 2455 2456 2457 2458 2459 2462
U 1198   2463 2465 2466 2467 2468 2469 2470 2471
```

```
:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2472 2477 2478 2479 2480 2481 2482 2483
U 11A8  2484 2485 2486 2487 2561 2562 2563 2564
U 11B0  2565 2566 2567 2568 2569 2570 2571 2572
U 11B8  2573 2574 2575 2576 2577 2578 2579 2580
U 11C0  2586 2589 2590 2591 2592 2594 2595 2596
U 11C8  2597 2598 2599 2600 2601 2602 2604 2605
U 11D0  2606 2607 2611 2612 2613 2614 2615 2616
U 11D8  2617 2618 2621 2622 2623 2624 2625 2626
U 11E0  2627 2631 2632 2635 2636 2637 2638 2639
U 11E8  2640 2641 2642 2643 2644 2645 2646 2664
U 11F0  2665 2666 2667 2670 2671 2672 2678 2679
U 11F8  2687 2693 2694 2697 2698 2699 2700 2701
U 1200  2704 2705 2707 2708 2709 2710 2711 2712
U 1208  2713 2714 2719 2720 2721 2722 2723 2724
U 1210  2725 2726 2727 2728 2731 2792 2793 2794
U 1218  2795 2796 2797 2798 2799 2800 2801 2802
U 1220  2803 2804 2805 2806 2807 2815 2816 2817
U 1228  2818 2819 2821 2822 2823 2824 2825 3520:
U 1230  2827 2828 2832 2833 2834 2844 2845 2846
U 1238  2847 2848 2849 2852 2865 2866 2867 2868
U 1240  2869 2870 2871 2872 3518: 2873 2891 2892
U 1248  2893 2894 2897 2898 2899 2902 3211 3212
U 1250  3213 3214 3215 3216 3217 3238 3239 3240
U 1258  3241 3294 3295 3296 3297 3298 3299 3300
U 1260  3311 3323 3324 3325 3326 3379 3380 3381
U 1268  3382 3383 3384 3385 3408 3409 3410 3411
U 1270  3464 3465 3466 3467 3468 3469 3470 3484
U 1278  3494 3495 3496 3497 3512 3519 3521 3523
U 1280 - 12A7 Unused
U 12A8    1966:
U 12B0 - 12BF Unused
U 12C0    3522:
U 12C8 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2739
ESKAR63.MCR                    MICRO2  1P(00)    26-JUL-85  17:08:26     Page   9

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR63        658

Used              658
Remaining

Total microwords used in memory U: 658
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR63.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents
   ESKAR64.MCR   MICRO2 1P(00)   26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2216 TEST 291 VERIFY DIVIDE STATES FOR FLOAT AND DOUBLE
; 2392 TEST 292 DIVIDE DOUBLE FLOATING ONE BY 0.1
; 2569 TEST 293 DIVIDE DOUBLE 0.1 BY FLOATING ONE
; 2767 TEST 294 DIVIDE DOUBLE FLOATING ONE BY FLOATING ONE
; 2914 TEST 295 DIVIDE DOUBLE WITH SELECTED OPERANDS
; 3108 TEST 296 TEST FOR NALU CARRY PROPAGATE FAULTS
; 3380 TEST 297 DIVIDE FLOAT FLOATING ONE BY FLOATING ONE
; 3526 TEST 298 RESERVED
; 3531 DUMMY NEWTST

```
;1 .NOLIST
;1804 .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR64
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844  ;*
            ;1845  ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846  ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847  ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848  ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849  ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850  ;
            ;1851  ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852  ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853  ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854  ; R[0F] AND DO A RETURN0.
            ;1855  ;
            ;1856  ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857  ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858  ; TO THE CONSOLE.
            ;1859  ;
            ;1860  ;-
U 1100.  0000.003C.1980.0800.0084.7003   ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101.  0000.003C.0580.0800.0084.7003   ;1862 1101: SC_K[.1],J/TRPH1  ; DATA UNALIGNED
U 1102.  0000.003C.0980.0800.0084.7003   ;1863 1102: SC_K[.2],J/TRPH1  ; X PAGE ERROR
U 1103.  0000.003C.0D80.0800.0084.7003   ;1864 1103: SC_K[.3],J/TRPH1  ; TB MODIFY
U 1104.  0000.003C.1180.0800.0084.7003   ;1865 1104: SC_K[.4],J/TRPH1  ; TB PROT ERROR
U 1105.  0000.003C.D580.0800.0084.7003   ;1866 1105: SC_K[.6],J/TRPH1  ; TB MISS
U 1106.  0000.003C.D980.0800.0084.7003   ;1867 1106: SC_K[.9],J/TRPH1  ; RES FLOAT OPER
U 1107.  0000.003C.5D80.0800.0084.7003   ;1868 1107: SC_K[.7],J/TRPH1  ; TB PARITY
U 1108.  0000.003C.0180.0800.0084.7003   ;1869 1108: SC_K[.8],J/TRPH1  ; CACHE PARITY
U 110C.  0000.003C.8580.0800.0084.7003   ;1870 110C: SC_K[.C],J/TRPH1  ; RDS
U 110D.  0000.003C.8980.0800.0084.7003   ;1871 110D: SC_K[.D],J/TRPH1  ; TIME OUT
U 110E.  0000.003C.6580.0800.0084.7003   ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F.  0000.003C.6180.0800.0084.7003   ;1873 110F: SC_K[.F],J/TRPH1  ; CS PARITY
U 1001.  0000.003C.81F0.2C00.0000.100A   ;1874 TRPH0: Q_ID[USTACK]
U 100A.  0C00.003C.01E0.0800.0000.100E   ;1875  Q_D,D_Q
U 100E.  0000.003C.8180.3C00.0000.101A   ;1876  ID[USTACK]_D
U 101A.  0C00.003C.01E0.0800.0000.101E   ;1877  Q_D,D_Q
U 101E.  0000.003C.01C0.0A78.0000.102A   ;1878  Q_R[0F]
U 102A.  0001.2024.0180.0800.0010.102E   ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102E.  0000.013C.0180.0800.0000.1002   ;1880  Z?     ; BRANCH IF NO
U 1002.  0001.2036.0180.0AF8.0000.0000   ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882  ;*
            ;1883  ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884  ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885  ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886  ;
            ;1887  ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888  ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889  ;       FAULT STATUS REGISTER
            ;1890  ;       SBI ERROR REGISTER
            ;1891  ;       TIMEOUT ADDRESS REGISTER
            ;1892  ;       MAINTENANCE REGISTER
            ;1893  ;       CACHE PARITY REGISTER
            ;1894  ;       TB ERROR REGISTER 1
            ;1895  ;       TB ERROR REGISTER 0
```

```
                ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1004   ;1897 TRPH1: RC[0D]_SC
U 1004. 0000.003D.D980.0800.0084.70ED   ;1898 =0 SETRCF[.9]
U 1005. 0000.003C.49F0.2C00.0000.1036   ;1899  Q_ID[TBER0]
U 1036. 0001.203C.4DF0.2DB0.0000.103E   ;1900  RC[6]_Q,Q_ID[TBER1]
U 103E. 0001.203C.65F0.2DB8.0000.1046   ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 1046. 0001.203C.6DF0.2DD8.0000.105B   ;1902  RC[0B]_Q,Q_ID[FAULT]
U 105B. 0001.203C.75F0.2DE0.0000.105F   ;1903  RC[0C]_Q,Q_ID[MAINT]
U 105F. 0001.203C.79F0.2DC8.0000.10D2   ;1904  RC[9]_Q,Q_ID[PARITY]
U 10D2. 0001.203C.69F0.2DC0.0000.10D3   ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 10D3. 0001.203C.81F0.2DD0.0000.1000   ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.10E7   ;1907 1000: RC[0E]_Q,ERROR1
                ;1908 ;*
                ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1911 ;  NEWTST
                ;1912 ;  ERLOOP
                ;1913 ;  ERROR1
                ;1914 ;  ERROR2
                ;1915 ;
U 10D4. 0000.003C.8580.0800.0084.70D5   ;1916 SCOPE: SC_K[.C]
U 10D5. 0803.001C.0180.0800.0000.1008   ;1917  ALU_NOT.MASK,D_ALU
U 1008. 0000.003D.7580.3C00.0000.1141   ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1009. 0000.003C.65F8.0800.0084.70D6   ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 10D6. 0818.0038.8D80.0800.0000.10D7   ;1920  D_K[.1F]    ; ...
U 10D7. 0D00.003C.0180.0800.0000.10D8   ;1921  D_DAL.SC
U 10D8. 0000.003C.3D80.3C00.0000.10D9   ;1922  ID[PSL]_D    ; WRITE THE PSL
U 10D9. 0818.0038.0580.0800.0000.10DA   ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 10DA. 0D00.003C.0180.0800.0000.10DB   ;1924  D_DAL.SC
U 10DB. 0F00.003C.3180.3C00.0000.10DC   ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10DC. 0000.003C.0180.0800.0000.10DD   ;1926  NOP
U 10DD. 0000.003C.3980.3C00.0000.10DE   ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 10DE. 0000.003C.2980.3C00.0000.10DF   ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 10DF. 0000.003C.4980.3C00.0000.10E0   ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 10E0. 0818.0038.C180.0800.0000.10E1   ;1930  D_K[.FFFF]
U 10E1. 0000.003C.6580.3C00.0000.10E2   ;1931  ID[SBI.ERR]_D
U 10E2. 0F00.003C.6D80.3C00.0000.10E3   ;1932  ID[FAULT]_D,D_0
U 10E3. 0000.003C.6D80.3C00.0000.10E4   ;1933  ID[FAULT]_D
U 10E4. 0000.003C.6580.3C00.0000.10E5   ;1934  ID[SBI.ERR]_D
U 10E5. 0003.003C.0180.0AF8.0000.105E   ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 10E6. 0818.0038.0980.0800.0000.105E   ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 10E7. 0810.0038.0180.0978.0000.10E8   ;1937 ERROR1: D_RC[0F]
U 10E8. 0819.0034.4D80.0800.0000.104E   ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E   ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 10E9. 0810.0038.0180.0978.0000.10EA   ;1940 ERROR2: D_RC[0F]
U 10EA. 0819.0034.4D80.0800.0000.105A   ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E   ;1942 105A: D_D.OR.K[.6],J/CALLCON
                ;1943 105E:
                ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E   ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
                ;1946 ;*
                ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
                ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1   ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2   ;1950 13F1: NOP
```

```
U 13F2, 0000.003C,0180,0800,0000,13F3  ;1951 13F2: NOP
U 13F3, 0000.003C,0180,0800,0000,13F4  ;1952 13F3: NOP
U 13F4, 0000.003C,0180,0800,0000,13F5  ;1953 13F4: NOP
U 13F5, 0000.003C,0180,0800,0000,13F6  ;1954 13F5: NOP
U 13F6, 0000.003C,0180,0800,0000,13F7  ;1955 13F6: NOP
U 13F7, 0000.003C,0180,0800,0000,13F8  ;1956 13F7: NOP
U 13F8, 0000.003C,0180,0800,0000,13F9  ;1957 13F8: NOP
U 13F9, 0000.003C,0180,0800,0000,13FA  ;1958 13F9: NOP
U 13FA, 0000.003C,0180,0800,0000,13FB  ;1959 13FA: NOP
U 13FB, 0000.003C,0180,0800,0000,13FC  ;1960 13FB: NOP
U 13FC, 0000.003C,0180,0800,0000,13FD  ;1961 13FC: NOP
U 13FD, 0000.003C,0180,0800,0000,13FE  ;1962 13FD: NOP
U 13FE, 0000.003C,0180,0800,0000,13FF  ;1963 13FE: NOP
U 13FF, 0000.003C,0180,0800,0000,1155  ;1964 13FF: NOP
U 1155, 0001.203E,59F0,2DE8,0000,0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001.203E,59F0,2DE8,0000,0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
           ;1967 ;*
           ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
           ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
           ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
           ;1971 ; TYPING IT.
           ;1972 ;-
U 10EB, 0810.0038,4D80,0978,0000,10EC  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 10EC, 0019.4016,4D80,09F8,0000,0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
           ;1975 ;*
           ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
           ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
           ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
           ;1979 ;-
U 10ED, 0010.0038,01C0,0978,0000,10EE  ;1980 SETRCF: Q_RC[0F]
U 10EE, 0019.2034,4DC0,0800,0000,10EF  ;1981  Q_Q.AND.K[.FF00]
U 10EF, 0019.2032,1D80,09F8,0000,0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
           ;1983 ;*****************************************************
           ;1984 ;
           ;1985 ; MAIN MACHINE/FPA SUBROUTINES
           ;1986 ;
           ;1987 ;*****************************************************
           ;1988
           ;1989 ; SHIFT D REGISTTER CONTENTS
           ;1990
           ;1991
           ;1992 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
           ;1993
           ;1994
U 10F0, 0000.003C,01F8,0800,0000,10F1  ;1995 SHIFTD: Q_0
U 10F1, 0500.003C,0180,0800,0000,10F2  ;1996  D_D.LEFT  ;
U 10F2, 0000.003C,05[illegible]0,0800,1414,B0F3  ;1997  EALU_STATE-K[.1],CLK.UBCC ;
U 10F3, 0000.123C,0180,0800,0000,100B  ;1998  EALU.Z?
U 100B, 0000.003C,0180,0800,0000,10F0  ;1999 =1011 J/SHIFTD  ;
U 100F, 0000.003E,0180,0800,0000,0001  ;2000  RETURN1
           ;2001
           ;2002 ;*
           ;2003 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
           ;2004 ; OR'S IN BIT 15, SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP TO
           ;2005 ; THAT ADDRESS.
```

```
            ;2006 ;-
            ;2007
U 10F4, 0819.0030.4580.0800.0000.10F5  ;2008 TRAP1: D_D.OR.K[.8000]
U 10F5, 0000.003C.6580.0800.1404.700C  ;2009  STATE_K[.10]
U 100C, 0000.003D.0180.0800.0000.10F0  ;2010 =0 CALL[SHIFTD]
U 100D, 0000.003C.5980.3C00.0000.10F6  ;2011  ID[ACC.2]_D
U 10F6, 0000.00FE.0380.0800.0000.0001  ;2012  TRAP.FPA,RETURN1
            ;2013
            ;2014 ;*
            ;2015 ; THIS ROUTINE GENERATES A UPC ADDRESS, AND JUMPS TO  THE
            ;2016 ; TRAP ROUTINE.
            ;2017 ;-
            ;2018
U 10F7, 0818.0038.4180.0800.0000.10F8  ;2019 TRP.89: D_K[.80]
U 10F8, 0000.003C.D980.0800.0000.10F9  ;2020  KMX/.9
U 10F9, 0819.0014.D980.0800.0000.10F4  ;2021  D_D*K[.9],J/TRAP1
            ;2022
            ;2023 ;*
            ;2024 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2025 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT
            ;2026 ; THE IB IS COMPLETELY FULL OF DATA.
            ;2027 ;
            ;2028 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2029 ;-
            ;2030
U 10FA, 3000.003C.0180.6000.0000.10FB  ;2031 IBLOAD: LOAD.IB,IBC/START
U 10FB, 0000.003C.0180.F800.0000.10FC  ;2032 SYNC02: MCT/ALLOW.IB.READ
U 10FC, 0000.003C.0180.F800.0000.10FD  ;2033  MCT/ALLOW.IB.READ
U 10FD, 0000.003C.0180.F800.0000.10FE  ;2034  MCT/ALLOW.IB.READ
U 10FE, 0000.003C.0180.F800.0000.10FF  ;2035  MCT/ALLOW.IB.READ
U 10FF, 0000.003C.0180.F800.0000.1109  ;2036  MCT/ALLOW.IB.READ
U 1109, 1000.003E.0180.F800.0000.0001  ;2037  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2038
            ;2039
            ;2040
            ;2041
            ;2042
            ;2043 ; THIS SUBROUTINE LOADS THE AR WITH DATA FOR THE RR AND NR,INITIATES
            ;2044 ; THE DIVIDE, AND READS BACK THE QUOTIENT HI, AND LO INTO THE Q AND D
            ;2045 ; REGISTERS, RESPECTIVELY. THE ROUTINE EXPECTS THE RR HI IN R6,THE
            ;2046 ; RR LO IN R7,THE NR HI IN R8, AND THE NR LO IN R9.
            ;2047
            ;2048 =0
U 1010, 0000.003D.0180.0800.0000.10F7  ;2049 LD.DIV: CALL[TRP.89]
U 1011, 0018.0038.4180.0AF0.0000.110A  ;2050  R[0E]_K[.80]
U 110A, 0000.003C.0180.0A70.0000.110B  ;2051  LAB_R[0E]  ; LA,LB GET BIT 7 ASSERTED
U 110B, 0000.007C.0180.0A70.0000.1110  ;2052  LAB_R[0E],CPSYNC ; FPA AT 089. INIT. MULT. SEQUENCER
U 1110, 0000.003C.0180.0A30.0000.1111  ;2053  LAB_R[6]  ; FPA AT 07F. ZERO REMAINDER REGISTER
U 1111, 0000.007C.0180.0A30.0000.1112  ;2054  LAB_R[6],CPSYNC  ; LOAD DIVIDEND HI
U 1112, 0000.003C.0180.0A38.0000.1113  ;2055  LAB_R[7]  ; FPA AT 0D6.
U 1113, 0000.007C.0180.0A38.0000.1114  ;2056  LAB_R[7],CPSYNC  ; LOAD DIVIDEND LO
U 1114, 0000.003C.0180.0800.0000.1115  ;2057  NOP  ; FPA AT 0D7. LOAD NR
U 1115, 0000.003C.0180.0A40.0000.1116  ;2058  LAB_R[8]  ; FPA AT 0D8
U 1116, 0000.007C.0180.0A40.0000.1117  ;2059  LAB_R[8],CPSYNC  ; LOAD DIVISOR HI
U 1117, 0000.003C.0180.0A48.0000.1118  ;2060  LAB_R[9]  ; FPA AT 0D9
```

```
U 1118, 0000,007C,0180,0A48,0000,1119  ;2061  LAB_R[9],CPSYNC  ; LOAD DIVISOR HI
U 1119, 0000,003C,0180,0800,0000,111A  ;2062  NOP   ; RR GETS DIVIDEND
        ;2063      ; NR GETS DIVISOR
U 111A, 0000,003C,0180,0800,0000,111B  ;2064  NOP   ; FPA AT 0DB. START DIVIDE
U 111B, 0000,063C,0180,0800,0000,1006  ;2065 WAIT: ACC.SYNC?  ; WAIT FOR DIVIDE DONE
U 1006, 0000,003C,0180,0800,0000,111B  ;2066 =110 J/WAIT   ;
U 1007, 0000,003C,01D8,0800,0000,111C  ;2067  Q_ACC   ; FPA AT 0D1.
        ;2068      ; SAVE QUOTIENT HI AND LO
        ;2069      ; IN AR LATCH AND IN
        ;2070      ; NORMALIZER REGISTER
        ;2071
U 111C, 0001,207C,0180,0988,0000,111D  ;2072  RC[1]_Q,CPSYNC  ; SAVE UNNORMALIZED QUOTIENT HI
U 111D, 0000,003C,01D8,0800,0000,111E  ;2073  Q_ACC   ; FPA AT 0D2
U 111E, 0001,207C,0180,0980,0000,111F  ;2074  RC[0]_Q,CPSYNC  ; SAVE UNNORMALIZED QUOTIENT LO
U 111F, 0000,003C,01D8,0800,0000,1120  ;2075  Q_ACC   ; FPA AT 0D3.
        ;2076      ; READ NORMALIZED QUOTIENT LO
U 1120, 0A00,003C,0180,0800,0000,1121  ;2077  D_ACC   ; FPA AT 0D4,
U 1121, 0000,003E,0180,0800,0000,0001  ;2078  RETURN1   ; READ NORMALIZED QUOTIENT HI
        ;2079
        ;2080
        ;2081 ;+
        ;2082 ; THIS ROUTINE TAKES A NUMBER FROM R0, WHICH INDICATES THE
        ;2083 ; PRESENT BIT POSITION OF THE FLOATING ONE PATTERN IN THE NR.
        ;2084 ; IT THEN GENERATES THE RESULTING QUOTIENT PATTERN WHEN RR IS
        ;2085 ; DIVIDED BY NR. THE QUOTIENT HI IS RETURNED IN THE Q REGISTER.
        ;2086 ; THE EXPECTED QUOTIENT LO IS RETURNED IN THE D REGISTER.
        ;2087 ;-
        ;2088
        ;2089 =0
U 1012, 0000,003C,01F8,0800,0000,1013  ;2090 DIV.CHK: Q_0
U 1013, 0F00,003C,0180,0800,0000,1122  ;2091  D_0
U 1122, 0018,0038,5580,0A88,0000,1123  ;2092  R[1]_K[.3F]
U 1123, 0000,003C,0180,0A00,0082,1124  ;2093 L0: SC_R[0]
U 1124, 0600,003C,0080,0800,0000,1125  ;2094 L1: D_D.RIGHT,SI/1
U 1125, 0000,003C,01B0,0800,0000,1126  ;2095  Q_Q.RIGHT,SI/ZERO
U 1126, 0000,003C,0580,0A08,0094,B127  ;2096  SC_SC-K[.1],CLK.UBCC,LAB_R[1]
U 1127, 0018,1200,0580,0A88,0000,101B  ;2097  EALU.Z?,R[1]_LA-K[.1]
U 101B, 0000,003C,0180,0800,0000,1124  ;2098 =1011 J/L1
U 101F, 0000,003C,0180,0A00,0082,1128  ;2099  SC_R[0]
U 1128, 0600,003C,0080,0800,0000,1129  ;2100 L2: D_D.RIGHT,SI/1
U 1129, 0000,003C,03B0,0800,0000,112A  ;2101  Q_Q.RIGHT,SI/7
U 112A, 0000,003C,0580,0A08,0094,B12B  ;2102  SC_SC-K[.1],CLK.UBCC,LAB_R[1]
U 112B, 0018,1200,0580,0A88,0000,102B  ;2103  EALU.Z?,R[1]_LA-K[.1]
U 102B, 0000,003C,0180,0800,0000,1128  ;2104 =1011 J/L2
U 102F, 0000,003C,0180,0A08,0010,112C  ;2105  ALU_R[1],CLK.UBCC
U 112C, 0000,1B3C,0180,0800,0000,1037  ;2106  ALU.N?
U 1037, 0000,003C,0180,0800,0000,1123  ;2107 =0111 J/L0
U 103F, 0600,003C,0080,0800,0000,112D  ;2108  D_D.RIGHT,SI/1
U 112D, 0000,003E,01B0,0800,0000,0001  ;2109  Q_Q.RIGHT,SI/ZERO,RETURN1
        ;2110
        ;2111 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRSSS IN THE D REGISTER AND
        ;2112 ; CONVERT IT TO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
        ;2113 ; THE FPA FOR A TRAP TO THAT ADDRESS.
        ;2114
U 112E, 0000,003C,65F8,0800,0084,712F  ;2115 GENTRA: SC_K[.10],Q_0
```

```
U 112F, 0D00.003C,8D80.0800.0084,7130  ;2116  D_DAL.SC,SC_K[.1F]
U 1130, 0801.001E,0180.0800.0000,0001  ;2117  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2118
            ;2119
            ;2120
            ;2121
            ;2122
            ;2123
            ;2124
            ;2125
            ;2126
            ;2127
            ;2128
            ;2129
            ;2130 ;
            ;2131 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
            ;2132 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
            ;2133 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
            ;2134 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
            ;2135 ;
            ;2136
U 1131, 0000.003C,01C0.0A28.0000,1132  ;2137 UNSEQA: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1132, 0800.003C,B980.0A20.0084,7133  ;2138  D_R[4],SC_K[.19]
U 1133, 0D00.003C,0180.0800.0000,1134  ;2139  D_DAL.SC
U 1134, 0000.003C,65E0.0800.0084,7135  ;2140  Q_D,SC_K[.10]
U 1135, 0D00.003C,0180.0800.0000,1136  ;2141  D_DAL.SC
U 1136, 0001.003C,0180.0AB8.0000,1137  ;2142  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 1137, 0800.003C,01C0.0A20.0000,1138  ;2143  D_R[4],Q_R[4]
U 1138, 0819.0024,5980.0800.0000,1139  ;2144  ALU_D[ANDNOT]K[.7F],D_ALU
U 1139, 0100.003C,5D88.0800.0084,713A  ;2145  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 113A, 0D00.003C,0180.0800.0000,113B  ;2146  D_DAL.SC
U 113B, 0001.003C,0180.0AB0.0084,713C  ;2147  R[6]_D,SC_K[.8]
U 113C, 0818.0034,59F8.0A28.0000,113D  ;2148  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 113D, 0D00.003C,01C0.0A30.0000,113E  ;2149  D_DAL.SC,Q_R[6]
U 113E, 001D.0032,0180.0AB0.0000,0001  ;2150  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
U 113F, 0D00.003C,01C0.0A30.0000,1140  ;2151  D_DAL.SC,Q_R[6]
U 1140, 001D.0032,0180.0AB0.0000,0001  ;2152  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
            ;2153
            ;2154 ;
            ;2155 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;2156 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;2157 ;   28: NAD/28   ; NOP
            ;2158 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;2159 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;2160 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;2161 ;  INITIALIZE ITSELF.
            ;2162 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;2163 ;
            ;2164
U 1141, 0818.0038,4580.0800.0000,1142  ;2165 FPINIT: D_K[.8000]
U 1142, 0000.003C,5D80.3C00.0000,1014  ;2166  ID[ACC.CS]_D   ; TURN ON FPA
U 1014, 0818.0039,2D80.0800.0000,112E  ;2167 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1015, 0000.003C,5980.3C00.0000,1143  ;2168  ID[ACC.2]_D
U 1143, 0000.00FC,0380.0800.0000,1016  ;2169  TRAP.FPA   ; TRAP FPA TO 28.
U 1016, 0018.0039,1980.0800.0200,1149  ;2170 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
```

```
U 1017. 0F00.003C.0180.0800.0000.1018  ;2171  D_0
U 1018. 0000.003D.0180.0800.0000.112E  ;2172 =0 CALL,J/GENTRA
U 1019. 0000.003C.5980.3C00.0000.1144  ;2173  ID[ACC.2]_D
U 1144. 0000.00FC.0380.0800.0000.101C  ;2174  TRAP.FPA  ; TRAP FPA TO 0
U 101C. 0818.0039.2D80.0800.0000.112E  ;2175 =0 D_K[.28],CALL,J/GENTRA
U 101D. 0000.003C.5980.3C00.0000.1145  ;2176  ID[ACC.2]_D
U 1145. 0000.00FE.0380.0800.0000.0001  ;2177  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
            ;2178 ;
            ;2179 ;
            ;2180 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2181 ;%0
            ;2182 ;$ 0000.0000, 0000.0000
            ;2183 ;
            ;2184 ;
            ;2185 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2186 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2187 ;
            ;2188
            ;2189 =0
U 1020. 0F00.003D.0180.0800.0000.112E  ;2190 FPIRD: D_0,CALL,J/GENTRA
U 1021. 0000.003C.5980.3C00.0000.1146  ;2191  ID[ACC.2]_D
U 1146. 0000.00FC.0380.0800.0000.1147  ;2192  TRAP.FPA  ; TRAP TO FPA 0
U 1147. 0000.003C.0180.0800.0000.1148  ;2193  NOP
U 1148. 0000.003E.0180.0800.0000.0001  ;2194  RETURN1
            ;2195 ;
            ;2196 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2197 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2198 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2199 ;
            ;2200
U 1149. 2000.003C.0180.0800.0000.114A  ;2201 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 114A. 3000.003C.0180.6000.0000.114B  ;2202  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 114B. 0000.003C.0180.F800.0000.114C  ;2203 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 114C. 0000.003C.0180.F800.0000.114D  ;2204  MCT/ALLOW.IB.READ
U 114D. 0000.003C.0180.F800.0000.114E  ;2205  MCT/ALLOW.IB.READ
U 114E. 0000.003C.0180.F800.0000.114F  ;2206  MCT/ALLOW.IB.READ
U 114F. 0000.003C.0180.F800.0000.1150  ;2207  MCT/ALLOW.IB.READ
U 1150. 1000.003C.0180.F800.0000.1151  ;2208  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1151. 0000.003E.0180.0800.0000.0001  ;2209  RETURN1
U 1152. 3000.003C.0180.0800.0000.114B  ;2210 WAITIB: IBC/START,J/IBW
            ;2211
            ;2212
            ;2213
            ;2214
            ;2215
```

```
;2216  .PAGE "TEST 291 VERIFY DIVIDE STATES FOR FLOAT AND DOUBLE"
;2217 ;*****************************************************************
;2218 ;+*
;2219 ; VERIFY DIVIDE STATES FOR FLOAT AND DOUBLE
;2220 ;
;2221 ; TEST DESCRIPTION
;2222 ;
;2223 ; THIS TEST DOES A SINGLE PRECISION  AND A DOUBLE
;2224 ; PRECISION DIVIDE. FIFTEEN STATES AFTER THE START
;2225 ; OF THE DIVIDE, THIRTY FOR DOUBLE PRECISION,THE FPA
;2226 ; MAINTENANCE REGISTER NEXT ADDRESS IS READ. THE NEXT
;2227 ; ADDRESS DETERMINES IF DIVIDE DONE HAS OCCURRED IN
;2228 ; THE REQUIRED NUMBER OF STATES. THIS TEST VERIFIES
;2229 ; THAT THE DIVIDE LOGIC ON THE FNM MODULE ,AND THE
;2230 ; TEMP/LSH REGISTERS ON THE FML MODULE CAN DEVELOP THE
;2231 ; QUOTIENT IN THE REQUIRED NUMBER OF STATES.
;2232 ;
;2233 ; FPA UCODE USED
;2234 ;
;2235 ; LOCATION CONTENTS
;2236 ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;2237 ; 07F: BSC/R,SCR,CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;2238 ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;2239 ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;2240 ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;2241 ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;2242 ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;2243 ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;2244 ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;2245 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2246 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2247 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2248 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2249 ;
;2250 ; LOGIC DESCRIPTION
;2251 ;
;2252 ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;2253 ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;2254 ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;2255 ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;2256 ; MODULE.
;2257 ;
;2258 ; ERROR DESCRIPTION
;2259 ;
;2260 ; EXPECTED NEXT ADDRESS
;2261 ; RECEIVED NEXT ADDRESS
;2262 ;
;2263 ; SYNC POINT
;2264 ;
;2265 ;--
;2266 ;
;2267 ;*****************************************************************
;2268 ;/////////////////////////////////////////////////////
;2269 ;+*
;2270 ; SUBTEST 1
```

```
            ;2271 ; TEST FOR PROPER NUMBER OF STATES FOR SINGLE PRECISION DIVIDE.
            ;2272 ;-
            ;2273 =0
U 1022. 0018.0039.0980.09F8.0000.10D4  ;2274 DIVT1: NEWTST[.2]
U 1023. 0000.003C.0180.0800.0000.1024  ;2275  NOP
U 1024. 0000.003D.0180.0800.0000.10EB  ;2276 =0 SUBTEST
U 1025. 0818.0038.2980.0800.0000.1153  ;2277  D_K[.34]
U 1153. 0100.003C.0180.0800.0000.1154  ;2278  D_D.LEFT2
U 1154. 0819.0030.0580.0800.0000.1156  ;2279  D_D.OR.K[.1]
U 1156. 0001.003C.0180.09F0.0000.1026  ;2280  RC[0E]_D  ; EXPECTED NAD 0D1
U 1026. 0000.003D.0180.0800.0000.10E6  ;2281 =0 ERLOOP
U 1027. 0018.0038.3180.0AA8.0000.1157  ;2282  R[5]_K[.40]  ; BIT 6 GETS PACKED TO BIT 61. (RR=0.11)
U 1157. 0003.003C.0180.0AB8.0000.1158  ;2283  R[7]_0   ; NR=0.1
U 1158. 0003.003C.0180.0AC0.0000.1159  ;2284  R[8]_0
U 1159. 0003.003C.0180.0AB0.0000.1028  ;2285  R[6]_0
U 1028. 2018.0039.0180.0800.4200.10FA  ;2286 =0 ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS DIVF R,R
U 1029. 0000.003C.0180.0800.0000.102C  ;2287  NOP
U 102C. 0000.003D.0180.0800.0000.10F7  ;2288 =0 CALL[TRP.89]  ;
U 102D. 0000.007C.0180.0800.0000.115A  ;2289  CPSYNC   ; FPA AT 089 . INIT. MULTIPLY SEQUENCER
U 115A. 0000.003C.0180.0A28.0000.115B  ;2290  LAB_R[5]  ; FPA AT 07F. ZERO REMAINDER REGISTER
U 115B. 0000.007C.0180.0A28.0000.115C  ;2291  LAB_R[5],CPSYNC  ; LOAD DIVIDEND HI
U 115C. 0000.003C.0180.0A30.0000.115D  ;2292  LAB_R[6]  ; FPA AT 0D6
U 115D. 0000.007C.0180.0A30.0000.115E  ;2293  LAB_R[6],CPSYNC  ; LOAD DIVIDEND LO
U 115E. 0000.003C.0180.0800.0000.115F  ;2294  NOP   ; FPA AT 0D7. LOAD NR
U 115F. 0000.003C.0180.0A38.0000.1160  ;2295  LAB_R[7]  ; FPA AT 0D8
U 1160. 0000.007C.0180.0A38.0000.1161  ;2296  LAB_R[7],CPSYNC  ; LOAD DIVISOR HI
U 1161. 0000.003C.0180.0A40.0000.1162  ;2297  LAB_R[8]  ; FPA AT 0D9
U 1162. 0000.007C.0180.0A40.0000.1163  ;2298  LAB_R[8],CPSYNC  ; LOAD DIVISOR LO
U 1163. 0000.003C.0180.0800.0000.1164  ;2299  NOP   ; FPA AT 0DA.
            ;2300     ; RR GETS DIVIDEND
            ;2301     ; NR GETS DIVISOR
U 1164. 0000.003C.0180.0800.0000.1165  ;2302  NOP   ; FPA AT 0D0. WAITING FOR DIVIDE DONE
U 1165. 0000.003C.0180.0800.0000.1166  ;2303  NOP   ; WAITING FOR DIVIDE DONE
U 1166. 0000.003C.0180.0800.0000.1167  ;2304  NOP
U 1167. 0000.003C.0180.0800.0000.1168  ;2305  NOP
U 1168. 0000.003C.0180.0800.0000.1169  ;2306  NOP
U 1169. 0000.003C.0180.0800.0000.116A  ;2307  NOP
U 116A. 0000.003C.0180.0800.0000.116B  ;2308  NOP
U 116B. 0000.003C.0180.0800.0000.116C  ;2309  NOP
U 116C. 0000.003C.0180.0800.0000.116D  ;2310  NOP
U 116D. 0000.003C.0180.0800.0000.116E  ;2311  NOP
U 116E. 0000.003C.0180.0800.0000.116F  ;2312  NOP
U 116F. 0000.003C.0180.0800.0000.1170  ;2313  NOP
U 1170. 0000.003C.0180.0800.0000.1171  ;2314  NOP   ; DIVIDE DONE WITH TEMP 29=1, 28=0
U 1171. 0000.003C.0180.0800.0000.1172  ;2315  NOP   ; DIVIDE DONE IN NEXT STATE
U 1172. 0000.003C.59F0.2C00.0000.1173  ;2316  Q_ID[ACC.2]  ; FPA AT 0D0. NAD=0D1
U 1173. 0019.2034.81C0.0800.0000.1174  ;2317  ALU_Q.AND.K[.3FF],Q_ALU
U 1174. 0001.203C.0180.09E8.0000.1175  ;2318  RC[0D]_Q  ; SAVE NAD
U 1175. 0011.2020.0180.0970.0010.1176  ;2319  ALU_Q.XOR.RC[0E],CLK.UBCC
U 1176. 0000.013C.0180.0800.0000.1030  ;2320  Z?
U 1030. 0000.003D.0180.0800.0000.10E7  ;2321 =0 ERROR1
U 1031. 0000.003C.0180.0800.0000.1032  ;2322  NOP
            ;2323 ;////////////////////////////////////////////////////
            ;2324 ;*
            ;2325 ; SUBTEST 2
```

```
            ;2326 ; TEST FOR PROPER NUMBER OF STATES FOR DOUBLE PRECISION DIVIDE.
            ;2327 ;-
U 1032. 0000.003D.0180.0800.0000.10EB  ;2328 =0 SUBTEST
U 1033. 0818.0038.2980.0800.0000.1177  ;2329  D_K[.34]
U 1177. 0100.003C.0180.0800.0000.1178  ;2330  D_D.LEFT2
U 1178. 0819.0030.0580.0800.0000.1179  ;2331  D_D.OR.K[.1]
U 1179. 0001.003C.0180.09F0.0000.1034  ;2332  RC[0E]_D  ; EXPECTED NAD 0D1.
            ;2333 =0
U 1034. 0000.003D.0180.0800.0000.10E6  ;2334  ERLOOP
U 1035. 0003.003C.0180.0AA8.0000.117A  ;2335  R[5]_0   ; RR=0.10
U 117A. 0003.003C.0180.0AB0.0000.117B  ;2336  R[6]_0
U 117B. 0018.0038.3180.0AB8.0000.117C  ;2337  R[7]_K[.40]  ; BIT 6 GETS PACKED TO BIT 61. (NR=0.11)
U 117C. 0003.003C.0180.0AC0.0000.1038  ;2338  R[8]_0
U 1038. 2018.0039.8580.0800.4200.10FA  ;2339 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; OPCODE IS DIVD R,R
U 1039. 0000.003C.0180.0800.0000.103A  ;2340  NOP
U 103A. 0000.003D.0180.0800.0000.10F7  ;2341 =0 CALL[TRP.89]  ;
U 103B. 0000.007C.0180.0800.0000.117D  ;2342  CPSYNC   ; FPA AT 089. INIT. MULTIPL SEQUENCER
U 117D. 0000.003C.0180.0A28.0000.117E  ;2343  LAB_R[5]  ; FPA AT 07F. ZERO REMAINDER REGISTER
U 117E. 0000.007C.0180.0A28.0000.117F  ;2344  LAB_R[5],CPSYNC  ; LOAD DIVIDEND HI
U 117F. 0000.003C.0180.0A30.0000.1180  ;2345  LAB_R[6]  ; FPA AT 0D6
U 1180. 0000.007C.0180.0A30.0000.1181  ;2346  LAB_R[6],CPSYNC  ; LOAD DIVIDEND LO
U 1181. 0000.003C.0180.0800.0000.1182  ;2347  NOP   ; FPA AT 0D7. LOAD NR
U 1182. 0000.003C.0180.0A38.0000.1183  ;2348  LAB_R[7]  ; FPA AT 0D8
U 1183. 0000.007C.0180.0A38.0000.1184  ;2349  LAB_R[7],CPSYNC  ; LOAD DIVISOR HI
U 1184. 0000.003C.0180.0A40.0000.1185  ;2350  LAB_R[8]  ; FPA AT 0D9
U 1185. 0000.007C.0180.0A40.0000.1186  ;2351  LAB_R[8],CPSYNC  ; LOAD DIVISOR LO
U 1186. 0000.003C.0180.0800.0000.1187  ;2352  NOP   ; FPA AT 0DA.
            ;2353     ; RR GETS DIVIDEND
            ;2354     ; NR GETS DIVISOR
U 1187. 0000.003C.0180.0800.0000.1188  ;2355  NOP   ; FPA AT 0DB. START DIVIDE
U 1188. 0000.003C.0180.0800.0000.1189  ;2356  NOP   ; FPA AT 0D0. WAITING FOR DIVIDE DONE
U 1189. 0000.003C.0180.0800.0000.118A  ;2357  NOP   ; WAITING FOR DIVIDE DONE
U 118A. 0000.003C.0180.0800.0000.118B  ;2358  NOP
U 118B. 0000.003C.0180.0800.0000.118C  ;2359  NOP
U 118C. 0000.003C.0180.0800.0000.118D  ;2360  NOP
U 118D. 0000.003C.0180.0800.0000.118E  ;2361  NOP
U 118E. 0000.003C.0180.0800.0000.118F  ;2362  NOP
U 118F. 0000.003C.0180.0800.0000.1190  ;2363  NOP
U 1190. 0000.003C.0180.0800.0000.1191  ;2364  NOP
U 1191. 0000.003C.0180.0800.0000.1192  ;2365  NOP
U 1192. 0000.003C.0180.0800.0000.1193  ;2366  NOP
U 1193. 0000.003C.0180.0800.0000.1194  ;2367  NOP
U 1194. 0000.003C.0180.0800.0000.1195  ;2368  NOP
U 1195. 0000.003C.0180.0800.0000.1196  ;2369  NOP
U 1196. 0000.003C.0180.0800.0000.1197  ;2370  NOP
U 1197. 0000.003C.0180.0800.0000.1198  ;2371  NOP
U 1198. 0000.003C.0180.0800.0000.1199  ;2372  NOP
U 1199. 0000.003C.0180.0800.0000.119A  ;2373  NOP
U 119A. 0000.003C.0180.0800.0000.119B  ;2374  NOP
U 119B. 0000.003C.0180.0800.0000.119C  ;2375  NOP
U 119C. 0000.003C.0180.0800.0000.119D  ;2376  NOP
U 119D. 0000.003C.0180.0800.0000.119E  ;2377  NOP
U 119E. 0000.003C.0180.0800.0000.119F  ;2378  NOP
U 119F. 0000.003C.0180.0800.0000.11A0  ;2379  NOP
U 11A0. 0000.003C.0180.0800.0000.11A1  ;2380  NOP
```

```
ZZ-ESKAR-V22  TEST 291 VERIFY DIVIDE STATES FOR FLOAT AND DOUBLE                SEQ 2755
ESKAR64.MCR                    MICRO2  1P(00)    26-JUL-85  17:09:33           Page   6

U 11A1. 0000.003C.0180.0800.0000.11A2  ;2381  NOP
U 11A2. 0000.003C.0180.0800.0000.11A3  ;2382  NOP
U 11A3. 0000.003C.0180.0800.0000.11A4  ;2383  NOP    ; DIVIDE DONE WITH TEMP 29=0, 28=1
U 11A4. 0000.003C.0180.0800.0000.11A5  ;2384  NOP    ; DIVIDE DONE IN THE NEXT STATE
U 11A5. 0000.003C.59F0.2C00.0000.11A6  ;2385  Q_ID[ACC.2]  ; FPA AT 0D0. NAD=0D1
U 11A6. 0019.2034.81C0.0800.0000.11A7  ;2386  ALU_Q.AND.K[.3FF],Q_ALU
U 11A7. 0001.203C.0180.09E8.0000.11A8  ;2387  RC[0D]_Q  ; SAVE NAD
U 11A8. 0011.2020.0180.0970.0010.11A9  ;2388  ALU_Q.XOR.RC[0E],CLK.UBCC
U 11A9. 0000.013C.0180.0800.0000.103C  ;2389  Z?
U 103C. 0000.003D.0180.0800.0000.10E7  ;2390 =0 ERROR1
U 103D. 0000.003C.0180.0800.0000.1040  ;2391  NOP
```

```
;2392  .PAGE "TEST 292 DIVIDE DOUBLE FLOATING ONE BY 0.1"
;2393 ;*************************************************************************
;2394 ;++
;2395 ; DIVIDE DOUBLE FLOATING ONE BY 0.1
;2396 ; TEST DESCRIPTION
;2397 ;
;2398 ; THIS TEST SUCCESSIVELY DIVIDES A FLOATING (1) PATTERN
;2399 ; IN THE REMAINDER REGISTER, WITH THE NORMALIZER REGISTER
;2400 ; CONTAINING ONLY THE HIDDEN BIT SET. THE RESULTING NORMALIZED
;2401 ; PACKED QUOTIENT IS IDENTICAL TO THE PATTERN IN THE REMAINDER
;2402 ; REGISTER.
;2403 ;
;2404 ; TEST SEQUENCE   DESCRIPTION
;2405 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2406 ; 2  PACK THE DATA
;2407 ; 3  PERFORM THE DIVIDE
;2408 ; 4  VERIFY RESULTS
;2409 ;
;2410 ; FPA UCODE USED
;2411 ;
;2412 ; LOCATION CONTENTS
;2413 ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;2414 ; 07F: BSC/R,SCR.CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;2415 ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;2416 ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;2417 ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;2418 ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;2419 ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;2420 ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;2421 ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;2422 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2423 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2424 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2425 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2426 ;
;2427 ; LOGIC DESCRIPTION
;2428 ;
;2429 ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;2430 ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;2431 ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;2432 ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;2433 ; MODULE.
;2434 ;
;2435 ; ERROR DESCRIPTION
;2436 ;
;2437 ; PACKED EXPECTED NORMALIZED QUOTIENT HI
;2438 ; PACKED RECEIVED NORMALIZED QUOTIENT HI
;2439 ; PACKED EXPECTED NORMALIZED QUOTIENT LO
;2440 ; PACKED RECEIVED NORMALIZED QUOTIENT LO
;2441 ; PACKED EXPECTED UNNORMALIZED QUOTIENT HI
;2442 ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
;2443 ; EXPECTED UNNORMALIZED QUOTIENT LO
;2444 ; RECEIVED UNNORMALIZED QUOTIENT LO
;2445 ; REMAINDER REGISTER HI
;2446 ; REMAINDER REGISTER LO
```

```
            ;2447 ; NORMALIZER REGISTER HI
            ;2448 ; NORMALIZER REGISTER LO
            ;2449 ; BIT COUNTER
            ;2450 ;
            ;2451 ;
            ;2452 ; SYNC POINT
            ;2453 ;
            ;2454 ;--
            ;2455 ;
            ;2456 ;*****************************************************************
            ;2457 ;/////////////////////////////////////////////////////
            ;2458 ;+
            ;2459 ; FLOAT A 1 THROUGH THE RR WITH NR=0.1, DIVIDE, CHECK RESULTS.
            ;2460 ;-
            ;2461 =0
U 1040, 0018.0039.8980.09F8.0000.10D4  ;2462 DIVT2: NEWTST[.D]
U 1041, 0818.0038.4D80.0800.0000.11AA  ;2463  D_K[.FF00]
U 11AA, 0819.0030.4180.0800.0000.11AB  ;2464  D_D.OR.K[.80]
U 11AB, 0001.0028.0180.0A88.0000.11AC  ;2465  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 11AC, 0000.003C.6180.0800.0084.71AD  ;2466  SC_K[.F]
U 11AD, 0000.003C.0580.0800.0084.B1AE  ;2467  SC_SC-K[.1]
U 11AE, 0801.0034.0180.0800.0000.11AF  ;2468  ALU_D.AND.MASK,D_ALU
U 11AF, 0001.0028.0180.0A90.0000.11B0  ;2469  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 11B0, 0818.0038.7980.0800.0000.11B1  ;2470  D_K[.30]
U 11B1, 0819.0030.8980.0800.0000.11B2  ;2471  D_D.OR.K[.D]
U 11B2, 0001.003C.0180.0990.0000.1042  ;2472  RC[2]_D   ; BIT COUNT OF 61
U 1042, 0000.003D.0180.0800.0000.10E6  ;2473 =0 ERLOOP
U 1043, 0810.0038.0180.0910.0082.11B3  ;2474 DIV1: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 11B3, 0019.0034.7580.0800.0010.11B4  ;2475  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING RR
            ;2476       ; LO WITH PATTERN, RR HI
            ;2477       ; WITH ZERO.
U 11B4, 0000.013C.0180.0800.0000.1044  ;2478  Z?
U 1044, 0000.003C.0180.0800.0000.11B7  ;2479 =0 J/DIV2
U 1045, 0803.001C.0180.0800.0000.11B5  ;2480  D_NOT.MASK  ; FLOATING 1
U 11B5, 0019.0024.5980.0AA8.0000.11B6  ;2481  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT BITS <6:0>
U 11B6, 0003.003C.0180.0AA0.0000.11B9  ;2482  R[4]_0,J/DIV3
U 11B7, 0003.003C.0180.0AA8.0000.11B8  ;2483 DIV2: R[5]_0   ; RR LO GETS 0
U 11B8, 0003.001C.0180.0AA0.0000.11B9  ;2484  R[4]_NOT.MASK  ; FLOATING 1
            ;2485 ; GENERATE UNNORMALIZED EXPECTED QUOTIENT
U 11B9, 0800.003C.0180.0A20.0000.11BA  ;2486 DIV3: D_R[4]
U 11BA, 0001.003C.0180.09B0.0000.11BB  ;2487  RC[6]_D   ; SAVE HI
U 11BB, 0800.003C.0180.0A28.0000.11BC  ;2488  D_R[5]
U 11BC, 0001.003C.0180.09A8.0000.11BD  ;2489  RC[5]_D   ; SAVE LO
U 11BD, 0000.003C.01C0.0A20.0000.11BE  ;2490  Q_R[4]
U 11BE, 0800.003C.0180.0A28.0000.11BF  ;2491  D_R[5]
U 11BF, 0000.003C.0128.0800.0000.11C0  ;2492  Q_Q.LEFT,SI/2
U 11C0, 0500.003C.0180.0800.0000.11C1  ;2493  D_D.LEFT,SI/ZERO
U 11C1, 0001.203C.0180.0AA0.0000.11C2  ;2494  R[4]_Q
U 11C2, 0001.003C.0180.0AA8.0000.1048  ;2495  R[5]_D
            ;2496 ; PACK THE DATA
U 1048, 0000.003D.0180.0800.0000.1131  ;2497 =0 CALL[UNSEQA]  ; PACK UNNORMALIZED EXPECTED QUOTIENT
U 1049, 0800.003C.0180.0A30.0000.11C3  ;2498  D_R[6]
U 11C3, 080D.0034.0180.0A10.0000.11C4  ;2499  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11C4, 0001.003C.0180.0AB0.0000.11C5  ;2500  R[6]_D
U 11C5, 0001.003C.0180.09D0.0000.11C6  ;2501  RC[0A]_D  ; SAVE
```

```
U 11C6, 0800,003C,0180,0A38,0000,11C7  ;2502  D_R[7]
U 11C7, 0001,003C,0180,09C0,0000,11C8  ;2503  RC[8]_D   ; SAVE
          ;2504 ; GENERATE NORMALIZED EXPECTED QUOTIENT
U 11C8, 0810,0038,0180,0930,0000,11C9  ;2505  D_RC[6]   ; NR HI
U 11C9, 0001,003C,0180,0AA0,0000,11CA  ;2506  R[4]_D
U 11CA, 0810,0038,0180,0928,0000,11CB  ;2507  D_RC[5]   ; NR LO
U 11CB, 0018,0038,31C0,0800,000C,11CC  ;2508  Q_K[.40]
U 11CC, 001D,2014,0180,0AA8,0010,11CD  ;2509  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 11CD, 0000,1B3C,0180,0800,0000,106E  ;2510  ALU.C?
U 106E, 0000,003C,0180,0800,0000,104A  ;2511 =1110 J/PAK1
U 106F, 0800,003C,0180,0A20,0000,11CE  ;2512  D_R[4]
U 11CE, 0019,0014,0580,0AA0,0000,104A  ;2513  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
          ;2514 ; PACK NORMALIZED EXPECTED QUOTIENT
          ;2515 =0
U 104A, 0000,003D,0180,0800,0000,1131  ;2516 PAK1: CALL[UNSEQA]  ; PACK NORMALIZED EXPECTED QUOTIENT
U 104B, 0800,003C,0180,0A30,0000,11CF  ;2517  D_R[6]
U 11CF, 080D,0034,0180,0A08,0000,11D0  ;2518  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11D0, 0001,003C,0180,0AB0,0000,11D1  ;2519  R[6]_D
U 11D1, 0001,003C,0180,09F0,0000,11D2  ;2520  RC[0E]_D  ; SAVE NORMALIZED EXPECTED QUOT. HI
U 11D2, 0800,003C,0180,0A38,0000,11D3  ;2521  D_R[7]
U 11D3, 0001,003C,0180,09E0,0000,11D4  ;2522  RC[0C]_D  ; SAVE NORMALIZED EXPECTED QUOT. LO
          ;2523 ; PACK OPERANDS FOR DIVIDE OPERATION
U 11D4, 0810,0038,0180,0930,0000,11D5  ;2524  D_RC[6]
U 11D5, 0001,003C,0180,0AA0,0000,11D6  ;2525  R[4]_D   ; RESTORE RR HI
U 11D6, 0810,0038,0180,0928,0000,11D7  ;2526  D_RC[5]
U 11D7, 0001,003C,0180,0AA8,0000,104C  ;2527  R[5]_D   ; RESTORE RR LO
U 104C, 0000,003D,0180,0800,0000,1131  ;2528 =0 CALL[UNSEQA]
U 104D, 0800,003C,0180,0A30,0000,11D8  ;2529  D_R[6]
U 11D8, 080D,0034,0180,0A08,0000,11D9  ;2530  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11D9, 0001,003C,0180,0AB0,0000,11DA  ;2531  R[6]_D  ; RR HI MASKED
U 11DA, 0003,003C,0180,0AC0,0000,11DB  ;2532  R[8]_0   ; NR HI
U 11DB, 0003,003C,0180,09A0,0000,11DC  ;2533  RC[4]_0   ; SAVE
U 11DC, 0003,003C,0180,0AC8,0000,11DD  ;2534  R[9]_0   ; NR LO
U 11DD, 0003,003C,0180,0998,0000,1050  ;2535  RC[3]_0   ; SAVE
U 1050, 2018,0039,8580,0800,4200,10FA  ;2536 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 1051, 0000,003C,0180,0800,0000,1052  ;2537  NOP
U 1052, 0000,003D,0180,0800,0000,1010  ;2538 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE
          ;2539     ; READ DATA INTO Q AND D.
U 1053, 0001,203C,0180,09D8,0000,11DE  ;2540  RC[0B]_Q  ; SAVE NORMALIZED QUOT. LO
U 11DE, 081C,2034,0180,0A08,0000,11DF  ;2541  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 11DF, 0001,003C,0180,09E8,0000,11E0  ;2542  RC[0D]_D  ; SAVE NORMALIZED QUOT. HI
U 11E0, 0000,003C,0180,0908,0000,11E1  ;2543  LC_RC[1]
U 11E1, 0810,0038,0180,0800,0000,11E2  ;2544  D_LC
U 11E2, 080D,0034,0180,0A10,0000,11E3  ;2545  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11E3, 0001,003C,0180,09C8,0000,11E4  ;2546  RC[9]_D   ; SAVE UNNORMALIZED QUOT. HI
U 11E4, 0000,003C,0180,0900,0000,11E5  ;2547  LC_RC[0]
U 11E5, 0010,0038,0180,09B8,0000,11E6  ;2548  RC[7]_LC  ; SAVE UNNORMALIZED QUOT. LO
U 11E6, 0011,2020,0180,0960,0010,11E7  ;2549  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED QUOTIENT LO
U 11E7, 0000,013C,0180,0800,0000,1054  ;2550  Z?
U 1054, 0000,003D,0180,0800,0000,10E7  ;2551 =0 ERROR1
U 1055, 0010,0038,01C0,0968,0000,11E8  ;2552  Q_RC[0D]
U 11E8, 0011,2020,0180,0970,0010,11E9  ;2553  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED QUOTIENT HI
U 11E9, 0000,013C,0180,0800,0000,1056  ;2554  Z?
U 1056, 0000,003D,0180,0800,0000,10E7  ;2555 =0 ERROR1
U 1057, 0010,0038,01C0,0940,0000,11EA  ;2556  Q_RC[8]
```

```
ZZ-ESKAR-V22  TEST 292 DIVIDE DOUBLE FLOATING ONE BY 0.1                                    SEQ 2759
ESKAR64.MCR                    MICRO2  1P(00)    26-JUL-85  17:09:33                      Page   6

U 11EA. 0011.2020.0180.0938.0010.11EB  ;2557  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT LO
U 11EB. 0000.013C.0180.0800.0000.1058  ;2558  Z?
U 1058. 0000.003D.0180.0800.0000.10E7  ;2559 =0 ERROR1
U 1059. 0010.0038.01C0.0950.0000.11EC  ;2560  Q_RC[0A]
U 11EC. 0011.2020.0180.0948.0010.11ED  ;2561  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT HI
U 11ED. 0000.013C.0180.0800.0000.105C  ;2562  Z?
U 105C. 0000.003D.0180.0800.0000.10E7  ;2563 =0 ERROR1
U 105D. 0810.0038.0180.0910.0010.11EE  ;2564  ALU_RC[2],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 11EE. 0000.013C.0180.0800.0000.1060  ;2565  Z?
U 1060. 0019.0000.0580.0990.0000.1043  ;2566 =0 ALU_D-K[.1],RC[2]_ALU,J/DIV1 ; DECREMENT BIT COUNTER
U 1061. 0000.003C.0180.0800.0000.1062  ;2567  NOP
            ;2568
```

```
;2569   .PAGE "TEST 293 DIVIDE DOUBLE 0.1 BY FLOATING ONE"
;2570  ;*****************************************************************
;2571  ;**
;2572  ; DIVIDE DOUBLE 0.1 BY FLOATING ONE
;2573  ;
;2574  ; TEST DESCRIPTION
;2575  ;
;2576  ; THIS TEST DIVIDES A FLOATING (1) PATTERN IN THE
;2577  ; NORMALIZER REGISTER INTO THE REMAINDER REGISTER. THE
;2578  ; REMAINDER REGISTER CONTAINS ONLY THE HIDDEN BIT SET.
;2579  ; THE QUOTIENT IS A REPEATING PATTERN OF ONES AND
;2580  ; ZEROS WHOSE CONFIGURATION CHANGES AS THE FLOATING
;2581  ; (1) PATTERN PROGRESSES THROUGH THE NR.
;2582  ;
;2583  ; TEST SEQUENCE   DESCRIPTION
;2584  ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2585  ; 2  PACK THE DATA
;2586  ; 3  PERFORM THE DIVIDE
;2587  ; 4  VERIFY RESULTS
;2588  ;
;2589  ; FPA UCODE USED
;2590  ;
;2591  ; LOCATION CONTENTS
;2592  ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;2593  ; 07F: BSC/R,SCR.CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;2594  ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;2595  ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;2596  ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;2597  ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;2598  ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;2599  ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;2600  ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;2601  ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2602  ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2603  ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2604  ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2605  ;
;2606  ; LOGIC DESCRIPTION
;2607  ;
;2608  ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;2609  ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;2610  ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;2611  ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;2612  ; MODULE.
;2613  ;
;2614  ; ERROR DESCRIPTION
;2615  ;
;2616  ; PACKED EXPECTED NORMALIZED QUOTIENT HI
;2617  ; PACKED RECEIVED NORMALIZED QUOTIENT HI
;2618  ; PACKED EXPECTED NORMALIZED QUOTIENT LO
;2619  ; PACKED RECEIVED NORMALIZED QUOTIENT LO
;2620  ; PACKED EXPECTED UNNORMALIZED QUOTIENT HI
;2621  ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
;2622  ; PACKED EXPECTED UNNORMALIZED QUOTIENT LO
;2623  ; PACKED RECEIVED UNNORMALIZED QUOTIENT LO
```

```
            ;2624 ; REMAINDER REGISTER HI
            ;2625 ; REMAINDER REGISTER LO
            ;2626 ; NORMALIZER REGISTER HI
            ;2627 ; NORMALIZER REGISTER LO
            ;2628 ; BIT COUNTER
            ;2629 ;
            ;2630 ;
            ;2631 ; SYNC POINT
            ;2632 ;
            ;2633 ;--
            ;2634 ;
            ;2635 ;*****************************************************************
            ;2636 ;/////////////////////////////////////////////////////
            ;2637 ;*
            ;2638 ; FLOAT A 1 THROUGH THE NR WITH RR=0.1, DIVIDE, CHECK RESULTS.
            ;2639 ;-
            ;2640 =0
U 1062. 0018.0039.8980.09F8.0000.10D4  ;2641 DIVT3: NEWTST[.D]
U 1063. 0818.0038.4D80.0800.0000.11EF  ;2642  D_K[.FF00]
U 11EF. 0819.0030.4180.0800.0000.11F0  ;2643  D_D.OR.K[.80]
U 11F0. 0001.0028.0180.0A98.0000.11F1  ;2644  R[3]_NOT.D  ; MASK FOR BITS <15:07>
U 11F1. 0000.003C.6180.0800.0084.71F2  ;2645  SC_K[.F]
U 11F2. 0000.003C.0580.0800.0084.B1F3  ;2646  SC_SC-K[.1]
U 11F3. 0801.0034.0180.0800.0000.11F4  ;2647  ALU_D.AND.MASK,D_ALU
U 11F4. 0001.0028.0180.0A90.0000.11F5  ;2648  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 11F5. 0818.0038.7980.0800.0000.11F6  ;2649  D_K[.30]
U 11F6. 0819.0030.8980.0800.0000.11F7  ;2650  D_D.OR.K[.D]
U 11F7. 0001.003C.0180.0990.0000.11F8  ;2651  RC[2]_D   ; BIT COUNT OF 61
U 11F8. 0003.003C.0180.0A80.0000.1064  ;2652  R[0]_0   ; INITIALIZE PATTERN COUNTER
U 1064. 0000.003D.0180.0800.0000.10E6  ;2653 =0 ERLOOP
U 1065. 0810.0038.0180.0910.0082.11F9  ;2654 DIV4: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 11F9. 0019.0034.7580.0800.0010.11FA  ;2655  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING RR
            ;2656      ; LO WITH PATTERN, RR HI
            ;2657      ; WITH ZERO.
U 11FA. 0000.013C.0180.0800.0000.1066  ;2658  Z?
U 1066. 0000.003C.0180.0800.0000.11FD  ;2659 =0 J/DIV5
U 1067. 0803.001C.0180.0800.0000.11FB  ;2660  D_NOT.MASK  ; FLOATING 1
U 11FB. 0019.0024.5980.0AA8.0000.11FC  ;2661  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT NR HI  BITS <6:0>
U 11FC. 0003.003C.D580.0AA0.0000.11FF  ;2662  R[4]_0,J/DIV6,KMX/.6 ; NR HI GETS 0
U 11FD. 0003.003C.D580.0AA8.0000.11FE  ;2663 DIV5: R[5]_0,KMX/.6  ; NR LO GETS 0
U 11FE. 0003.001C.0180.0AA0.0000.11FF  ;2664  R[4]_NOT.MASK  ; FLOATING 1
U 11FF. 0810.0038.0180.0910.0000.1200  ;2665 DIV6: D_RC[2]
U 1200. 0019.0000.D580.0800.0010.1201  ;2666  ALU_D-K[.6],CLK.UBCC
U 1201. 0000.013C.0180.0800.0000.1068  ;2667  Z?
U 1068. 0000.003C.0180.0800.0000.1207  ;2668 =0 J/DIV6A   ; NOT AT BIT 6
U 1069. 0003.003C.0180.0998.0000.1202  ;2669  RC[3]_0   ; AT BIT 6
U 1202. 0003.003C.0180.09A0.0000.1203  ;2670  RC[4]_0
U 1203. 0003.003C.0180.0AD0.0000.1204  ;2671  R[0A]_0
U 1204. 0003.003C.0180.0AD8.0000.1205  ;2672  R[0B]_0
U 1205. 0003.003C.0180.09F0.0000.1206  ;2673  RC[0E]_0
U 1206. 0003.003C.0180.09E0.0000.106C  ;2674  RC[0C]_0,J/DIV6C  ;
U 1207. 0000.1B3C.0180.0800.0000.1047  ;2675 DIV6A: ALU.N?
U 1047. 0000.003C.0180.0800.0000.1208  ;2676 =0111 J/DIV6B
U 104F. 0000.003C.0180.0800.0000.106C  ;2677  J/DIV6C
            ;2678
```

```
            ;2679 ; GENERATE EXPECTED QUOTIENT
U 1208. 0800.003C.0180.0A20.0000.1209  ;2680 DIV6B: D_R[4]
U 1209. 0001.003C.0180.09A0.0000.120A  ;2681  RC[4]_D   ; SAVE HI
U 120A. 0800.003C.0180.0A28.0000.120B  ;2682  D_R[5]
U 120B. 0001.003C.0180.0998.0000.120C  ;2683  RC[3]_D   ; SAVE LO
U 120C. 0000.003C.0180.0A00.0000.120D  ;2684  LAB_R[0]
U 120D. 0018.0014.0580.0A80.0000.106A  ;2685  R[0]_LA+K[.1]  ; UPDATE PATTERN COUNTER
U 106A. 0000.003D.0180.0800.0000.1012  ;2686 =0 CALL[DIV.CHK]  ;
U 106B. 0001.203C.0180.0AD0.0000.120E  ;2687  R[0A]_Q   ; SAVE QUOTIENT HI
U 120E. 0001.003C.0180.0AD8.0000.120F  ;2688  R[0B]_D   ; SAVE QUOTIENT LO
U 120F. 0001.203C.0180.0AA0.0000.1210  ;2689  R[4]_Q
U 1210. 0001.003C.0180.0AA8.0000.106C  ;2690  R[5]_D
            ;2691 ; PACK THE UNNORMALIZED EXPECTED QUOTIENT
            ;2692 =0
U 106C. 0000.003D.0180.0800.0000.1131  ;2693 DIV6C: CALL[UNSEQA]  ;
U 106D. 0800.003C.0180.0A30.0000.1211  ;2694  D_R[6]
U 1211. 080D.0034.0180.0A18.0000.1212  ;2695  ALU_D[AND]R[3],D_ALU ; MASK OUT BITS <15:7>
U 1212. 0001.003C.0180.0AB0.0000.1213  ;2696  R[6]_D
U 1213. 0001.003C.0180.09D0.0000.1214  ;2697  RC[0A]_D  ; SAVE
U 1214. 0800.003C.0180.0A38.0000.1215  ;2698  D_R[7]
U 1215. 0001.003C.0180.09C0.0000.1216  ;2699  RC[8]_D   ; SAVE
            ;2700 ; GENERATE NORMALIZED EXPECTED QUOTIENT
U 1216. 0800.003C.0180.0A50.0000.1217  ;2701  D_R[0A]
U 1217. 0001.003C.0180.0AA0.0000.1218  ;2702  R[4]_D   ; RESTORE QUOTIENT HI
U 1218. 0800.003C.0180.0A58.0000.1219  ;2703  D_R[0B]
U 1219. 0001.003C.0180.0AA8.0000.121A  ;2704  R[5]_D   ; RESTORE QUOTIENT LO
U 121A. 0018.0038.31C0.0800.0000.121B  ;2705  Q_K[.40]
U 121B. 001D.2014.0180.0AA8.0010.121C  ;2706  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 121C. 0000.1B3C.0180.0800.0000.107E  ;2707  ALU.C?
U 107E. 0000.003C.0180.0800.0000.1070  ;2708 =1110 J/PAK2
U 107F. 0800.003C.0180.0A20.0000.121D  ;2709  D_R[4]
U 121D. 0019.0014.0580.0AA0.0000.1070  ;2710  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
            ;2711 ; PACK THE NORMALIZED EXPECTED QUOTIENT
            ;2712 =0
U 1070. 0000.003D.0180.0800.0000.1131  ;2713 PAK2: CALL[UNSEQA]  ;
U 1071. 0800.003C.0180.0A30.0000.121E  ;2714  D_R[6]
U 121E. 080D.0034.0180.0A18.0000.121F  ;2715  ALU_D[AND]R[3],D_ALU ; MASK OUT BITS <15:7>
U 121F. 0001.003C.0180.0AB0.0000.1220  ;2716  R[6]_D
U 1220. 0001.003C.0180.09F0.0000.1221  ;2717  RC[0E]_D  ; SAVE NORMALIZED EXPECTED QUOT. HI
U 1221. 0800.003C.0180.0A38.0000.1222  ;2718  D_R[7]
U 1222. 0001.003C.0180.09E0.0000.1223  ;2719  RC[0C]_D  ; SAVE NORMALIZED EXPECTED QUOT. LO
            ;2720 ; PACK OPERANDS FOR DIVIDE OPERATION
U 1223. 0810.0038.0180.0920.0000.1224  ;2721  D_RC[4]
U 1224. 0001.003C.0180.0AA0.0000.1225  ;2722  R[4]_D   ; RESTORE NR HI
U 1225. 0810.0038.0180.0918.0000.1226  ;2723  D_RC[3]
U 1226. 0001.003C.0180.0AA8.0000.1072  ;2724  R[5]_D   ; RESTORE NR LO
U 1072. 0000.003D.0180.0800.0000.1131  ;2725 =0 CALL[UNSEQA]
U 1073. 0800.003C.0180.0A30.0000.1227  ;2726  D_R[6]
U 1227. 080D.0034.0180.0A18.0000.1228  ;2727  ALU_D[AND]R[3],D_ALU ; MASK OUT BITS <15:7>
U 1228. 0001.003C.0180.0AC0.0000.1229  ;2728  R[8]_D   ; HOLD NR HI FOR DIVIDE
U 1229. 0800.003C.0180.0A38.0000.122A  ;2729  D_R[7]
U 122A. 0001.003C.0180.0AC8.0000.122B  ;2730  R[9]_D   ; HOLD NR LO FOR DIVIDE
U 122B. 0003.003C.0180.09B0.0000.122C  ;2731  RC[6]_0   ; SAVE RR HI
U 122C. 0003.003C.0180.09A8.0000.122D  ;2732  RC[5]_0   ; SAVE RR LO
U 122D. 0003.003C.0180.0AB0.0000.122E  ;2733  R[6]_0   ; HOLD RR HI FOR DIVIDE
```

```
U 122E, 0003,003C,0180,0AB8,0000,1074  ;2734  R[7]_0   ; HOLD RR LO FOR DIVIDE
U 1074, 2018,0039,8580,0800,4200,10FA  ;2735 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 1075, 0000,003C,0180,0800,0000,1076  ;2736  NOP
U 1076, 0000,003D,0180,0800,0000,1010  ;2737 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE
      ;2738     ; READ DATA INTO Q AND D.
U 1077, 0001,203C,0180,09D8,0000,122F  ;2739  RC[0B]_Q  ; SAVE NORMALIZED QUOT. LO
U 122F, 081C,2034,0180,0A18,0000,1230  ;2740  D_D.AND.R[3]  ; MASK OUT BITS<15:07>
U 1230, 0001,003C,0180,09E8,0000,1231  ;2741  RC[0D]_D  ; SAVE NORMALIZED QUOT. HI
U 1231, 0000,003C,0180,0908,0000,1232  ;2742  LC_RC[1]
U 1232, 0810,0038,0180,0800,0000,1233  ;2743  D_LC
U 1233, 080D,0034,0180,0A18,0000,1234  ;2744  ALU_D[AND]R[3],D_ALU ; MASK OUT BITS <15:7>
U 1234, 0001,003C,0180,09C8,0000,1235  ;2745  RC[9]_D   ; SAVE UNNORMALIZED QUOT. HI
U 1235, 0000,003C,0180,0900,0000,1236  ;2746  LC_RC[0]
U 1236, 0010,0038,0180,09B8,0000,1237  ;2747  RC[7]_LC  ; SAVE UNNORMALIZED QUOT. LO
U 1237, 0011,2020,0180,0960,0010,1238  ;2748  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED QUOTIENT LO
U 1238, 0000,013C,0180,0800,0000,1078  ;2749  Z?
U 1078, 0000,003D,0180,0800,0000,10E7  ;2750 =0 ERROR1
U 1079, 0010,0038,01C0,0968,0000,1239  ;2751  Q_RC[0D]
U 1239, 0011,2020,0180,0970,0010,123A  ;2752  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED QUOTIENT HI
U 123A, 0000,013C,0180,0800,0000,107A  ;2753  Z?
U 107A, 0000,003D,0180,0800,0000,10E7  ;2754 =0 ERROR1
U 107B, 0010,0038,01C0,0940,0000,123B  ;2755  Q_RC[8]
U 123B, 0011,2020,0180,0938,0010,123C  ;2756  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT LO
U 123C, 0000,013C,0180,0800,0000,107C  ;2757  Z?
U 107C, 0000,003D,0180,0800,0000,10E7  ;2758 =0 ERROR1
U 107D, 0010,0038,01C0,0950,0000,123D  ;2759  Q_RC[0A]
U 123D, 0011,2020,0180,0948,0010,123E  ;2760  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT HI
U 123E, 0000,013C,0180,0800,0000,1080  ;2761  Z?
U 1080, 0000,003D,0180,0800,0000,10E7  ;2762 =0 ERROR1
U 1081, 0810,0038,0180,0910,0010,123F  ;2763  ALU_RC[2],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 123F, 0000,013C,0180,0800,0000,1082  ;2764  Z?
U 1082, 0019,0000,0580,0990,0000,1065  ;2765 =0 ALU_D-K[.1],RC[2]_ALU,J/DIV4 ; DECREMENT BIT COUNTER
U 1083, 0000,003C,0180,0800,0000,1084  ;2766  NOP
```

```
;2767   .PAGE "TEST 294 DIVIDE DOUBLE FLOATING ONE BY FLOATING ONE"
;2768  ;*****************************************************************
;2769  ;**
;2770  ; DIVIDE DOUBLE FLOATING ONE BY FLOATING ONE
;2771  ;
;2772  ; TEST DESCRIPTION
;2773  ;
;2774  ; THIS TEST DIVIDES A FLOATING (1) PATTERN IN THE
;2775  ; REMAINDER REGISTER BY A FLOATING (1) PATTERN
;2776  ; IN THE NORMALIZER REGISTER. THIS PATTERN AUGMENTS
;2777  ; THE THE FAULT DETECTION ACCOMPLISHED BY THE PREVIOUS
;2778  ; TESTS.
;2779  ;
;2780  ; TEST SEQUENCE   DESCRIPTION
;2781  ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2782  ; 2  PACK THE DATA
;2783  ; 3  PERFORM THE DIVIDE
;2784  ; 4  VERIFY RESULTS
;2785  ;
;2786  ; FPA UCODE USED
;2787  ;
;2788  ; LOCATION CONTENTS
;2789  ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;2790  ; 07F: BSC/R.SCR.CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;2791  ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;2792  ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;2793  ; 0D8: BSC/R.SCR/CPU,FADC/AR,NAD/0D9,WAIT
;2794  ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;2795  ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;2796  ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;2797  ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;2798  ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2799  ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2800  ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2801  ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2802  ;
;2803  ; LOGIC DESCRIPTION
;2804  ;
;2805  ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;2806  ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;2807  ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;2808  ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;2809  ;
;2810  ; ERROR DESCRIPTION
;2811  ;
;2812  ; PACKED EXPECTED NORMALIZED QUOTIENT HI
;2813  ; PACKED RECEIVED NORMALIZED QUOTIENT HI
;2814  ; PACKED EXPECTED NORMALIZED QUOTIENT LO
;2815  ; PACKED RECEIVED NORMALIZED QUOTIENT LO
;2816  ; PACKED EXPECTED UNNORMALIZED QUOTIENT HI
;2817  ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
;2818  ; PACKED EXPECTED UNNORMALIZED QUOTIENT LO
;2819  ; PACKED RECEIVED UNNORMALIZED QUOTIENT LO
;2820  ; REMAINDER REGISTER HI
;2821  ; REMAINDER REGISTER LO
```

```
            ;2822 ; NORMALIZER REGISTER HI
            ;2823 ; NORMALIZER REGISTER LO
            ;2824 ; BIT COUNTER
            ;2825 ;
            ;2826 ; SYNC POINT
            ;2827 ;
            ;2828 ;--
            ;2829 ;
            ;2830 ;*************************************************************************
            ;2831 ;/////////////////////////////////////////////////////
            ;2832 ;*
            ;2833 ; FLOAT A1 THROUGH THE RR AND NR. DIVIDE AND CHECK RESULTS.
            ;2834 ;-
            ;2835 =0
U 1084. 0018,0039,8980,09F8,0000,10D4  ;2836 DIVT4: NEWTST[.D]
U 1085. 0818,0038,4D80,0800,0000,1240  ;2837  D_K[.FF00]
U 1240. 0819,0030,4180,0800,0000,1241  ;2838  D_D.OR.K[.80]
U 1241. 0001,0028,0180,0A88,0000,1242  ;2839  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 1242. 0000,003C,6180,0800,0084,7243  ;2840  SC_K[.F]
U 1243. 0000,003C,0580,0800,0084,B244  ;2841  SC_SC-K[.1]
U 1244. 0801,0034,0180,0800,0000,1245  ;2842  ALU_D.AND.MASK,D_ALU
U 1245. 0001,0028,0180,0A90,0000,1246  ;2843  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 1246. 0818,0038,7980,0800,0000,1247  ;2844  D_K[.30]
U 1247. 0819,0030,8980,0800,0000,1248  ;2845  D_D.OR.K[.D]
U 1248. 0001,003C,0180,0990,0000,1086  ;2846  RC[2]_D   ; BIT COUNT OF 61
U 1086. 0000,003D,0180,0800,0000,10E6  ;2847 =0 ERLOOP
U 1087. 0810,0038,0180,0910,0082,1249  ;2848 DIV7: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 1249. 0019,0034,7580,0800,0010,124A  ;2849  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING RR
            ;2850       ; LO WITH PATTERN, RR HI
            ;2851       ; WITH ZERO.
U 124A. 0000,013C,0180,0800,0000,1088  ;2852  Z?
U 1088. 0000,003C,0180,0800,0000,124D  ;2853 =0 J/DIV8
U 1089. 0803,001C,0180,0800,0000,124B  ;2854  D_NOT.MASK  ; FLOATING 1
U 124B. 0019,0024,5980,0AA8,0000,124C  ;2855  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT BITS <6:0>
U 124C. 0003,003C,0180,0AA0,0000,1255  ;2856  R[4]_0,J/DIV9
U 124D. 0003,003C,0180,0AA8,0000,124E  ;2857 DIV8: R[5]_0   ; RR LO GETS 0
U 124E. 0003,001C,0180,0AA0,0000,124F  ;2858  R[4]_NOT.MASK  ; FLOATING 1
U 124F. 0800,003C,0180,0A20,0000,1250  ;2859  D_R[4]
U 1250. 0001,003C,0180,09B0,0000,1251  ;2860  RC[6]_D   ; SAVE RR HI
U 1251. 0001,003C,0180,09A0,0000,1252  ;2861  RC[4]_D   ; SAVE NR HI
U 1252. 0800,003C,0180,0A28,0000,1253  ;2862  D_R[5]
U 1253. 0001,003C,0180,09A8,0000,1254  ;2863  RC[5]_D   ; SAVE RR LO
U 1254. 0001,003C,0180,0998,0000,1255  ;2864  RC[3]_D   ; SAVE NR LO
U 1255. 0003,003C,0180,09F0,0000,1256  ;2865 DIV9: RC[0E]_0  ; EXPECTED NORMALIZED QUOTIENT HI
U 1256. 0003,003C,0180,09E0,0000,1257  ;2866  RC[0C]_0  ; EXPECTED  NORMALIZED QUOTIENT LO
U 1257. 0003,003C,0180,09D0,0000,1258  ;2867  RC[0A]_0  ; EXPECTED UNNORMALIZED QUOTIENT HI
U 1258. 0003,003C,0180,09C0,0000,108A  ;2868  RC[8]_0   ; EXPECTED UNNORMALIZED QUOTIENT LO
            ;2869 ; PACK OPERANDS FOR DIVIDE OPERATION
U 108A. 0000,003D,0180,0800,0000,1131  ;2870 =0 CALL[UNSEQA]
U 108B. 0800,003C,0180,0A30,0000,1259  ;2871  D_R[6]
U 1259. 080D,0034,0180,0A08,0000,125A  ;2872  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 125A. 0001,003C,0180,0AB0,0000,125B  ;2873  R[6]_D   ; HOLD RR HI MASKED FOR DIVIDE
U 125B. 0001,003C,0180,0AC0,0000,125C  ;2874  R[8]_D   ; HOLD NR HI MASKED FOR DIVIDE
U 125C. 0800,003C,0180,0A38,0000,125D  ;2875  D_R[7]   ;
U 125D. 0001,003C,0180,0AC8,0000,108C  ;2876  R[9]_D   ; HOLD NR LO FOR DIVIDE
```

```
U 108C. 2018.0039.8580.0800.4200.10FA  ;2877 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 108D. 0000.003C.0180.0800.0000.108E  ;2878  NOP
U 108E. 0000.003D.0180.0800.0000.1010  ;2879 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE
        ;2880     ; READ DATA INTO Q AND D.
U 108F. 0001.203C.0180.09D8.0000.125E  ;2881  RC[0B]_Q  ; SAVE NORMALIZED QUOT. LO
U 125E. 081C.2034.0180.0A08.0000.125F  ;2882  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 125F. 0001.003C.0180.09E8.0000.1260  ;2883  RC[0D]_D  ; SAVE NORMALIZED QUOT. HI
U 1260. 0000.003C.0180.0908.0000.1261  ;2884  LC_RC[1]
U 1261. 0810.0038.0180.0800.0000.1262  ;2885  D_LC
U 1262. 080D.0034.0180.0A10.0000.1263  ;2886  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 1263. 0001.003C.0180.09C8.0000.1264  ;2887  RC[9]_D  ; SAVE UNNORMALIZED QUOT. HI
U 1264. 0000.003C.0180.0900.0000.1265  ;2888  LC_RC[0]
U 1265. 0010.0038.0180.09B8.0000.1266  ;2889  RC[7]_LC  ; SAVE UNNORMALIZED QUOT. LO
U 1266. 0011.2020.0180.0960.0010.1267  ;2890  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED QUOTIENT LO
U 1267. 0000.013C.0180.0800.0000.1090  ;2891  Z?
U 1090. 0000.003D.0180.0800.0000.10E7  ;2892 =0 ERROR1
U 1091. 0010.0038.01C0.0968.0000.1268  ;2893  Q_RC[0D]
U 1268. 0011.2020.0180.0970.0010.1269  ;2894  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED QUOTIENT HI
U 1269. 0000.013C.0180.0800.0000.1092  ;2895  Z?
U 1092. 0000.003D.0180.0800.0000.10E7  ;2896 =0 ERROR1
U 1093. 0010.0038.01C0.0940.0000.126A  ;2897  Q_RC[8]
U 126A. 0011.2020.0180.0938.0010.126B  ;2898  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT LO
U 126B. 0000.013C.0180.0800.0000.1094  ;2899  Z?
U 1094. 0000.003D.0180.0800.0000.10E7  ;2900 =0 ERROR1
U 1095. 0010.0038.01C0.0950.0000.126C  ;2901  Q_RC[0A]
U 126C. 0011.2020.0180.0948.0010.126D  ;2902  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT HI
U 126D. 0000.013C.0180.0800.0000.1096  ;2903  Z?
U 1096. 0000.003D.0180.0800.0000.10E7  ;2904 =0 ERROR1
U 1097. 0810.0038.0180.0910.0010.126E  ;2905  ALU_RC[2],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 126E. 0000.013C.0180.0800.0000.1098  ;2906  Z?
U 1098. 0019.0000.0580.0990.0000.1087  ;2907 =0 ALU_D-K[.1],RC[2]_ALU,J/DIV7 ; DECREMENT BIT COUNTER
U 1099. 0000.003C.0180.0800.0000.109A  ;2908  NOP
        ;2909 ; OPCODE DATA
        ;2910 ;* 8
        ;2911 ;$ 5546. 0056
        ;2912 ;$ 5566. 0056
        ;2913
```

```
;2914  .PAGE "TEST 295 DIVIDE DOUBLE WITH SELECTED OPERANDS"
;2915 ;*****************************************************************
;2916 ;**
;2917 ; DIVIDE DOUBLE WITH SELECTED OPERANDS
;2918 ;
;2919 ; TEST DESCRIPTION
;2920 ;
;2921 ; THIS TEST USES A SELECTED GROUP OF DIVIDE PATTERNS.
;2922 ; THESE PATTERNS SUPPLEMENT THE PREVIOUS DIVIDE TEST
;2923 ; PATTERNS.
;2924 ;
;2925 ; TEST SEQUENCE   DESCRIPTION
;2926 ;
;2927 ; 1  INITIALIZE WORD COUNT AND STARTING ADDRESS
;2928 ; 2  FETCH THE DIVIDEND AND DIVISOR
;2929 ; 3  PERFORM THE DIVIDE
;2930 ; 4  VERIFY RESULTS
;2931 ;
;2932 ; FPA UCODE USED
;2933 ;
;2934 ; LOCATION CONTENTS
;2935 ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;2936 ; 07F: BSC/R,SCR.CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;2937 ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;2938 ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;2939 ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;2940 ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;2941 ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;2942 ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;2943 ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;2944 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2945 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2946 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2947 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2948 ;
;2949 ; LOGIC DESCRIPTION
;2950 ;
;2951 ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;2952 ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;2953 ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;2954 ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;2955 ; MODULE.
;2956 ;
;2957 ; ERROR DESCRIPTION
;2958 ;
;2959 ; PACKED REMAINDER REGISTER HI
;2960 ; PACKED REMAINDER REGISTER LO
;2961 ; PACKED NORMALIZER REGISTER HI
;2962 ; PACKED NORMALIZER REGISTER LO
;2963 ; PACKED EXPECTED NORMALIZED QUOTIENT HI
;2964 ; PACKED RECEIVED NORMALIZED QUOTIENT HI
;2965 ; PACKED EXPECTED NORMALIZED QUOTIENT LO
;2966 ; PACKED RECEIVED NORMALIZED QUOTIENT LO
;2967 ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
;2968 ; PACKED RECEIVED UNNORMALIZED QUOTIENT LO
```

```
            :2969 ;
            :2970 ;
            :2971 ; SYNC POINT
            :2972 ;
            :2973 ;--
            :2974 ;
            :2975 ;*****************************************************************
            :2976 ;/////////////////////////////////////////////////////////////////
            :2977 ;*
            :2978 ; DIVIDE OPERATION WITH SELECTED OPERANDS
            :2979 ;-
            :2980 =0
U 109A. 0018.0039.8980.09F8.0000.10D4  :2981 DIVT5: NEWTST[.D]
U 109B. 0818.0038.4980.0800.0000.126F  :2982  D_K[.FF]
U 126F. 0819.0000.0580.0800.0000.1270  :2983  D_D-K[.1]
U 1270. 0500.003C.0180.0800.0000.1271  :2984  D_D.LEFT  ; WORD COUNT ADDRESS 1FC
U 1271. 0001.003C.0180.0800.0200.1272  :2985  VA_D
U 1272. 0000.003C.0180.4000.0000.1273  :2986  D[LONG]_CACHE
U 1273. 0001.003C.0180.0AE0.0000.1274  :2987  R[0C]_D   ; SAVE WORD COUNT
U 1274. 0818.0038.7580.0800.0000.1275  :2988  D_K[.20]
U 1275. 0100.003C.0180.0800.0000.1276  :2989  D_D.LEFT2
U 1276. 0100.003C.0180.0800.0000.1277  :2990  D_D.LEFT2
U 1277. 0001.003C.0180.0AE8.0000.1278  :2991  R[0D]_D   ; FIRST ADDRESS IS 200
U 1278. 0818.0038.4D80.0800.0000.1279  :2992  D_K[.FF00]
U 1279. 0819.0030.4180.0800.0000.127A  :2993  D_D.OR.K[.80]
U 127A. 0001.0028.0180.09A0.0000.109C  :2994  RC[4]_NOT.D  ; MASK OUT BITS <15:7>
U 109C. 0000.003D.0180.0800.0000.10E6  :2995 =0 ERLOOP
            :2996 ; FETCH DIVIDEND AND DIVISOR
U 109D. 0800.003C.0180.0A68.0000.127B  :2997 DIVP: D_R[0D]  ; SAVE
U 127B. 0001.003C.0180.0800.0200.127C  :2998  VA_D   ; ADDRESS
U 127C. 0000.003C.0180.4000.0000.127D  :2999  D[LONG]_CACHE
U 127D. 0811.0034.0180.0920.0000.127E  :3000  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS, <15:7>
U 127E. 0001.003C.0180.09F0.0000.127F  :3001  RC[0E]_D  ; SAVE RR HI
U 127F. 0001.003C.0180.0AB0.0000.1280  :3002  R[6]_D   ; HOLD FOR DIVIDE
U 1280. 0000.003C.0180.0803.0000.1281  :3003  VA_VA+4   ; ADDRESS
U 1281. 0000.003C.0180.4000.0000.1282  :3004  D[LONG]_CACHE
U 1282. 0001.003C.0180.0AB8.0000.1283  :3005  R[7]_D   ; HOLD FOR DIVIDE
U 1283. 0001.003C.0180.09E8.0000.1284  :3006  RC[0D]_D  ; SAVE RR LO
U 1284. 0000.003C.0180.0803.0000.1285  :3007  VA_VA+4   ; ADDRESS
U 1285. 0000.003C.0180.4000.0000.1286  :3008  D[LONG]_CACHE  ;
U 1286. 0811.0034.0180.0920.0000.1287  :3009  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 1287. 0001.003C.0180.0AC0.0000.1288  :3010  R[8]_D   ; HOLD FOR DIVIDE
U 1288. 0001.003C.0180.09E0.0000.1289  :3011  RC[0C]_D  ; SAVE NR HI
U 1289. 0000.003C.0180.0803.0000.128A  :3012  VA_VA+4   ; ADDRESS
U 128A. 0000.003C.0180.0801.0000.128B  :3013  PC_VA   ; SAVE
U 128B. 0000.003C.0180.4000.0000.128C  :3014  D[LONG]_CACHE  ;
U 128C. 0001.003C.0180.0AC8.0000.128D  :3015  R[9]_D   ; HOLD FOR DIVIDE
U 128D. 0001.003C.0180.09D8.0000.109E  :3016  RC[0B]_D  ; SAVE NR LO
U 109E. 2018.0039.8580.0800.4200.10FA  :3017 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 109F. 0000.003C.0180.0800.0000.10A0  :3018  NOP
            :3019 ; PERFORM THE DIVIDE
U 10A0. 0000.003D.0180.0800.0000.1010  :3020 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE,
            :3021      ; READ DATA INTO Q AND D
U 10A1. 0001.203C.0180.09B8.0000.128E  :3022  RC[7]_Q   ; SAVE NORMALIZED QUOTIENT LO
U 128E. 0811.0034.0180.0920.0000.128F  :3023  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
```

```
U 128F, 0001.003C.0180.09C8.0000.1290  ;3024  RC[9]_D   ; SAVE NORMALIZED QUOTIENT HI
U 1290, 0810.0038.0180.0908.0000.1291  ;3025  D_RC[1]
U 1291, 0001.003C.0180.09B0.0000.1292  ;3026  RC[6]_D   ; SAVE UNNORMALIZED QUOTIENT HI
U 1292, 0810.0038.0180.0900.0000.1293  ;3027  D_RC[0]
U 1293, 0001.003C.0180.09A8.0000.1294  ;3028  RC[5]_D   ; SAVE UNNORMALIZED QUOTIENT LO
U 1294, 0014.0038.0180.0800.0200.1295  ;3029  VA_PC    ; RESTORE VA
U 1295, 0000.003C.0180.0803.0000.1296  ;3030  VA_VA+4   ; ADDRESS
U 1296, 0000.003C.0180.4000.0000.1297  ;3031  D[LONG]_CACHE  ; EXPECTED NORMALIZED QUOTIENT HI
U 1297, 0811.0034.0180.0920.0000.1298  ;3032  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 1298, 0001.003C.0180.09D0.0000.1299  ;3033  RC[0A]_D  ; SAVE
U 1299, 0010.0038.01C0.0948.0000.129A  ;3034  Q_RC[9]
                ;3035 ; VERIFY RESULTS
U 129A, 0011.2020.0180.0950.0010.129B  ;3036  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK QUOT. HI
U 129B, 0000.013C.0180.0800.0000.10A2  ;3037  Z?
U 10A2, 0000.003D.0180.0800.0000.10E7  ;3038 =0 ERROR1
U 10A3, 0000.003C.0180.0803.0000.129C  ;3039  VA_VA+4   ; ADDRESS
U 129C, 0000.003C.0180.4000.0000.129D  ;3040  D[LONG]_CACHE  ; EXPECTED NORMALIZED QUOTIENT LO
U 129D, 0001.003C.0180.09C0.0000.129E  ;3041  RC[8]_D   ; SAVE
U 129E, 0010.0038.01C0.0938.0000.129F  ;3042  Q_RC[7]
U 129F, 0011.2020.0180.0940.0010.12A0  ;3043  ALU_Q.XOR.RC[8],CLK.UBCC ; CHECK QUOT. LO
U 12A0, 0000.013C.0180.0800.0000.10A4  ;3044  Z?
U 10A4, 0000.003D.0180.0800.0000.10E7  ;3045 =0 ERROR1
U 10A5, 0800.003C.D580.0A60.0000.12A1  ;3046  D_R[0C],KMX/.6  ; CHECK WORD COUNT
U 12A1, 0019.0000.D580.0AE0.0010.12A2  ;3047  ALU_D-K[.6],R[0C]_ALU,CLK.UBCC ; DECREMENT WORD COUNT
U 12A2, 0000.013C.0180.0800.0000.10A6  ;3048  Z?
U 10A6, 0000.003C.7D80.0800.0000.12A3  ;3049 =0 J/DIVP1,KMX/.18
U 10A7, 0000.003C.0180.0800.0000.12A5  ;3050  J/DIVP2
U 12A3, 0800.003C.0180.0A68.0000.12A4  ;3051 DIVP1: D_R[0D]
U 12A4, 0019.0014.7D80.0AE8.0000.109D  ;3052  ALU_D+K[.18],R[0D]_ALU,J/DIVP ; UPDATE PATTERN ADDRESS COUNTER
U 12A5, 0000.003C.0180.0800.0000.10A8  ;3053 DIVP2: NOP
            ;3054 ; TABLE OF DIVIDE PATTERNS
            ;3055 ; WORD COUNT
            ;3056 ;% 1FC
            ;3057 ;$ 002A, 0000
            ;3058 ;% 200
            ;3059 ; DIVIDE PATTERN 1
            ;3060 ;$ 43F6, 0000
            ;3061 ;$ 0000, 0000
            ;3062 ;$ 0000, 0000
            ;3063 ;$ 0000, 0000
            ;3064 ;$ 0076, 0000
            ;3065 ;$ 0000, 0000
            ;3066 ; DIVIDE PATTERN 2
            ;3067 ;$ 4078, 51EB
            ;3068 ;$ 851E, B852
            ;3069 ;$ 3FA3, D70A
            ;3070 ;$ 3D70, A3D7
            ;3071 ;$ 0042, 0000
            ;3072 ;$ 0000, 0000
            ;3073 ; DIVIDE PATTERN 3
            ;3074 ;$ 47FF, FE00
            ;3075 ;$ 0000, 0000
            ;3076 ;$ 4428, 0000
            ;3077 ;$ 0000, 0000
            ;3078 ;$ 0043, 0AAA
```

```
;3079 ;$ AAAA, AAAB
;3080 ; DIVIDE PATTERN 4
;3081 ;$ 43FC, 0000
;3082 ;$ 0000, 0000
;3083 ;$ 43FE, 0000
;3084 ;$ 0000, 0000
;3085 ;$ 007D, FBF7
;3086 ;$ EFDF, BF7F
;3087 ; DIVIDE PATTERN 5
;3088 ;$ 4187, BC28
;3089 ;$ F5C2, 8F5C
;3090 ;$ BEE1, 47AE
;3091 ;$ 147A, E148
;3092 ;$ 001A, 3E8B
;3093 ;$ A2E8, BA2E
;3094 ; DIVIDE PATTERN 6
;3095 ;$ 600A, C723
;3096 ;$ 0489, E800
;3097 ;$ 2293, 92EE
;3098 ;$ 8E92, 1D5D
;3099 ;$ 00F0, BDC2
;3100 ;$ 1ABB, 48DB
;3101 ; DIVIDE PATTERN 7
;3102 ;$ 2293, 92EE
;3103 ;$ 8E92, 1D5D
;3104 ;$ 630A, C723
;3105 ;$ 0489, E800
;3106 ;$ 0008, 1CEA
;3107 ;$ 1454, 5C75
```

```
;3108  .PAGE "TEST 296 TEST FOR NALU CARRY PROPAGATE FAULTS"
;3109 ;*****************************************************************
;3110 ;**
;3111 ; TEST FOR NALU CARRY PROPAGATE FAULTS
;3112 ;
;3113 ; TEST DESCRIPTION
;3114 ;
;3115 ; THE REMAINDER MOST SIGNIFICANT (8) BITS AND THE
;3116 ; NORMALIZER REGISTER (8) MOST SIGNIFICANT BITS ARE
;3117 ; SELECTED SO THAT WHEN DIVIDED, A CARRY GENERATE
;3118 ; FROM THE LOW (4) BITS WILL PROPAGATE THROUGH
;3119 ; THE HIGH (4) BITS. THIS PATTERN IS THEN SHIFTED
;3120 ; RIGHT (4) BITS AND THE TEST IS REPEATED. THIS
;3121 ; SEQUENCE CONTINUES UNTILL THE PATTERN RESIDES IN
;3122 ; THE LEAST SIGNIFICANT BYTE OF BOTH REGISTERS.
;3123 ; CARRY PROAGATE FAULTS ARE DETECTED BY THIS
;3124 ; PATTERN.
;3125 ;
;3126 ; TEST SEQUENCE   DESCRIPTION
;3127 ;
;3128 ; 1  INITIALIZE WORD COUNT AND STARTING ADDRESS
;3129 ; 2  FETCH THE DIVIDEND AND DIVISOR
;3130 ; 3  PERFORM THE DIVIDE
;3131 ; 4  VERIFY RESULTS
;3132 ;
;3133 ; FPA UCODE USED
;3134 ;
;3135 ; LOCATION CONTENTS
;3136 ; 089: MSC/IR0.OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;3137 ; 07F: BSC/R,SCR.CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;3138 ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;3139 ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;3140 ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;3141 ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;3142 ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;3143 ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;3144 ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;3145 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;3146 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;3147 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;3148 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;3149 ;
;3150 ; LOGIC DESCRIPTION
;3151 ;
;3152 ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;3153 ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;3154 ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;3155 ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;3156 ; MODULE.
;3157 ;
;3158 ; ERROR DESCRIPTION
;3159 ;
;3160 ; PACKED REMAINDER REGISTER HI
;3161 ; PACKED REMAINDER REGISTER LO
;3162 ; PACKED NORMALIZER REGISTER HI
```

```
            ;3163 ; PACKED NORMALIZER REGISTER LO
            ;3164 ; PACKED EXPECTED NORMALIZED QUOTIENT HI
            ;3165 ; PACKED RECEIVED NORMALIZED QUOTIENT HI
            ;3166 ; PACKED EXPECTED NORMALIZED QUOTIENT LO
            ;3167 ; PACKED RECEIVED NORMALIZED QUOTIENT LO
            ;3168 ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
            ;3169 ; PACKED RECEIVED UNNORMALIZED QUOTIENT LO
            ;3170 ;
            ;3171 ;
            ;3172 ; SYNC POINT
            ;3173 ;
            ;3174 ;--
            ;3175 ;
            ;3176 ;*****************************************************************************
            ;3177 =0
U 10A8, 0018.0039.F580.09F8.0000.10D4  ;3178 DIVT6: NEWTST[.A]
U 10A9, 0818.0038.9580.0800.0000.12A6  ;3179  D_K[.B0]
U 12A6, 0819.0030.6180.0800.0000.12A7  ;3180  D_D.OR.K[.F]
U 12A7, 0100.003C.0180.0800.0000.12A8  ;3181  D_D.LEFT2  ; WORD COUNT ADDRESS 2FC
U 12A8, 0001.003C.0180.0800.0200.12A9  ;3182  VA_D
U 12A9, 0000.003C.0180.4000.0000.12AB  ;3183  D[LONG]_CACHE
U 12AB, 0001.003C.0180.0AE0.0000.12AC  ;3184  R[0C]_D   ; SAVE WORD COUNT
U 12AC, 0818.0038.C980.0800.0000.12AD  ;3185  D_K[.3030]
U 12AD, 0200.003C.0180.0800.0000.12AE  ;3186  D_D.RIGHT2
U 12AE, 0200.003C.0180.0800.0000.12AF  ;3187  D_D.RIGHT2
U 12AF, 0019.0024.6180.0AE8.0000.12B0  ;3188  R[0D]_ALU,ALU_D.ANDNOT.K[.F] ; FIRST ADDRESS IS 300
U 12B0, 0818.0038.4D80.0800.0000.12B1  ;3189  D_K[.FF00]
U 12B1, 0819.0030.4180.0800.0000.12B2  ;3190  D_D.OR.K[.80]
U 12B2, 0001.0028.0180.09A0.0000.10AA  ;3191  RC[4]_NOT.D  ; MASK OUT BITS <15:7>
U 10AA, 0000.003D.0180.0800.0000.10E6  ;3192 =0 ERLOOP
            ;3193 ; FETCH DIVIDEND AND DIVISOR
U 10AB, 0800.003C.0180.0A68.0000.12B3  ;3194 DIVP3: D_R[0D]   ; SAVE
U 12B3, 0001.003C.0180.0800.0200.12B4  ;3195  VA_D   ; ADDRESS
U 12B4, 0000.003C.0180.4000.0000.12B5  ;3196  D[LONG]_CACHE
U 12B5, 0001.003C.0180.0AA0.0000.12B6  ;3197  R[4]_D   ; RR HI
U 12B6, 0001.003C.0180.09F0.0000.12B7  ;3198  RC[0E]_D  ; SAVE RR HI
U 12B7, 0000.003C.0180.0803.0000.12B8  ;3199  VA_VA+4   ; ADDRESS
U 12B8, 0000.003C.0180.4000.0000.12B9  ;3200  D[LONG]_CACHE
U 12B9, 0001.003C.0180.0AA8.0000.12BA  ;3201  R[5]_D   ; RR LO
U 12BA, 0001.003C.0180.09E8.0000.10AC  ;3202  RC[0D]_D  ; SAVE RR LO
U 10AC, 0000.003D.0180.0800.0000.1131  ;3203 =0 CALL[UNSEQA]
U 10AD, 0800.003C.0180.0A30.0000.12BB  ;3204  D_R[6]
U 12BB, 0811.0034.0180.0920.0000.12BC  ;3205  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS, <15:7>
U 12BC, 0001.003C.0180.0AB0.0000.12BD  ;3206  R[6]_D   ; HOLD FOR DIVIDE
U 12BD, 0001.003C.0180.0998.0000.12BE  ;3207  RC[3]_D   ; HOLD RR HI PACKED
U 12BE, 0800.003C.0180.0A38.0000.12BF  ;3208  D_R[7]
U 12BF, 0001.003C.0180.0990.0000.12C0  ;3209  RC[2]_D   ; HOLD RR LO PACKED
U 12C0, 0000.003C.0180.0803.0000.12C1  ;3210  VA_VA+4   ; ADDRESS
U 12C1, 0000.003C.0180.4000.0000.12C2  ;3211  D[LONG]_CACHE  ;
U 12C2, 0001.003C.0180.0AA0.0000.12C3  ;3212  R[4]_D   ; NR HI
U 12C3, 0001.003C.0180.09E0.0000.12C4  ;3213  RC[0C]_D  ; SAVE NR HI
U 12C4, 0000.003C.0180.0803.0000.12C5  ;3214  VA_VA+4   ; ADDRESS
U 12C5, 0000.003C.0180.0801.0000.12C6  ;3215  PC_VA   ; SAVE
U 12C6, 0000.003C.0180.4000.0000.12C7  ;3216  D[LONG]_CACHE  ;
U 12C7, 0001.003C.0180.0AA8.0000.12C8  ;3217  R[5]_D   ; NR LO
```

```
U 12C8. 0001.003C.0180.09D8.0000.10AE  ;3218  RC[0B]_D  ; SAVE NR LO
U 10AE. 0000.003D.0180.0800.0000.1131  ;3219 =0 CALL[UNSEQA]
U 10AF. 0800.003C.0180.0A30.0000.12C9  ;3220  D_R[6]
U 12C9. 0811.0034.0180.0920.0000.12CA  ;3221  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS, <15:7>
U 12CA. 0001.003C.0180.0AC0.0000.12CB  ;3222  R[8]_D   ; HOLD NR HI FOR DIVIDE
U 12CB. 0800.003C.0180.0A38.0000.12CC  ;3223  D_R[7]
U 12CC. 0001.003C.0180.0AC8.0000.12CD  ;3224  R[9]_D   ; HOLD NR LO FOR DIVIDE
U 12CD. 0810.0038.0180.0970.0000.12CE  ;3225  D_RC[0E]
U 12CE. 0001.003C.0180.0AB0.0000.12CF  ;3226  R[6]_D   ; HOLD RR HI FOR DIVIDE
U 12CF. 0810.0038.0180.0968.0000.12D0  ;3227  D_RC[0D]
U 12D0. 0001.003C.0180.0AB8.0000.10B0  ;3228  R[7]_D   ; HOLD RR LO FOR DIVIDE
U 10B0. 2018.0039.8580.0800.4200.10FA  ;3229 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 10B1. 0000.003C.0180.0800.0000.10B2  ;3230  NOP
          ;3231 ; PERFORM THE DIVIDE
U 10B2. 0000.003D.0180.0800.0000.1010  ;3232 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE,
          ;3233      ; READ DATA INTO Q AND D
U 10B3. 0001.203C.0180.09B8.0000.12D1  ;3234  RC[7]_Q   ; SAVE NORMALIZED QUOTIENT LO
U 12D1. 0811.0034.0180.0920.0000.12D2  ;3235  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 12D2. 0001.003C.0180.09C8.0000.12D3  ;3236  RC[9]_D   ; SAVE NORMALIZED QUOTIENT HI
U 12D3. 0810.0038.0180.0908.0000.12D4  ;3237  D_RC[1]
U 12D4. 0001.003C.0180.09B0.0000.12D5  ;3238  RC[6]_D   ; SAVE UNNORMALIZED QUOTIENT HI
U 12D5. 0810.0038.0180.0900.0000.12D6  ;3239  D_RC[0]
U 12D6. 0001.003C.0180.09A8.0000.12D7  ;3240  RC[5]_D   ; SAVE UNNORMALIZED QUOTIENT LO
U 12D7. 0014.0038.0180.0800.0200.12D8  ;3241  VA_PC    ; RESTORE VA
U 12D8. 0000.003C.0180.0803.0000.12D9  ;3242  VA_VA+4   ; ADDRESS
U 12D9. 0000.003C.0180.4000.0000.12DA  ;3243  D[LONG]_CACHE  ; EXPECTED NORMALIZED QUOTIENT HI
U 12DA. 0811.0034.0180.0920.0000.12DB  ;3244  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 12DB. 0001.003C.0180.09D0.0000.12DC  ;3245  RC[0A]_D  ; SAVE
U 12DC. 0010.0038.01C0.0948.0000.12DD  ;3246  Q_RC[9]
          ;3247 ; VERIFY RESULTS
U 12DD. 0011.2020.0180.0950.0010.12DE  ;3248  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK QUOT. HI
U 12DE. 0000.013C.0180.0800.0000.10B4  ;3249  Z?
U 10B4. 0000.003D.0180.0800.0000.10E7  ;3250 =0 ERROR1
U 10B5. 0000.003C.0180.0803.0000.12DF  ;3251  VA_VA+4   ; ADDRESS
U 12DF. 0000.003C.0180.4000.0000.12E0  ;3252  D[LONG]_CACHE  ; EXPECTED NORMALIZED QUOTIENT LO
U 12E0. 0001.003C.0180.09C0.0000.12E1  ;3253  RC[8]_D   ; SAVE
U 12E1. 0010.0038.01C0.0938.0000.12E2  ;3254  Q_RC[7]
U 12E2. 0011.2020.0180.0940.0010.12E3  ;3255  ALU_Q.XOR.RC[8],CLK.UBCC ; CHECK QUOT. LO
U 12E3. 0000.013C.0180.0800.0000.10B6  ;3256  Z?
U 10B6. 0000.003D.0180.0800.0000.10E7  ;3257 =0 ERROR1
U 10B7. 0800.003C.D580.0A60.0000.12E4  ;3258  D_R[0C],KMX/.6  ; CHECK WORD COUNT
U 12E4. 0019.0000.D580.0AE0.0010.12E5  ;3259  ALU_D-K[.6],R[0C]_ALU,CLK.UBCC ; DECREMENT WORD COUNT
U 12E5. 0000.013C.0180.0800.0000.10B8  ;3260  Z?
U 10B8. 0000.003C.7D80.0800.0000.12E6  ;3261 =0 J/DIVP4,KMX/.18
U 10B9. 0000.003C.0180.0800.0000.12E8  ;3262  J/DIVP5
U 12E6. 0800.003C.0180.0A68.0000.12E7  ;3263 DIVP4: D_R[0D]
U 12E7. 0019.0014.7D80.0AE8.0000.10AB  ;3264  ALU_D+K[.18],R[0D]_ALU,J/DIVP3 ; UPDATE PATTERN ADDRESS COUNTER
U 12E8. 0000.003C.0180.0800.0000.10BA  ;3265 DIVP5: NOP
          ;3266
          ;3267
          ;3268 ; TABLE OF DIVIDE PATTERNS
          ;3269 ; WORD COUNT
          ;3270 ;% 2FC
          ;3271 ;$ 005A, 0000
          ;3272 ;% 300
```

```
;3273 ; PATTERN 1
;3274 ;$ 0000. 1800
;3275 ;$ 0000. 0000
;3276 ;$ 0000. 1700
;3277 ;$ 0000. 0000
;3278 ;$ 003C. 75B3
;3279 ;$ 7E87. 5B38
;3280 ; PATTERN 2
;3281 ;$ 0000. 1580
;3282 ;$ 0000. 0000
;3283 ;$ 0000. 1570
;3284 ;$ 0000. 0000
;3285 ;$ 003F. E448
;3286 ;$ A94B. 1888
;3287 ; PATTERN 3
;3288 ;$ 0000. 1558
;3289 ;$ 0000. 0000
;3290 ;$ 0000. 1557
;3291 ;$ 0000. 0000
;3292 ;$ 0040. 1C43
;3293 ;$ 72AE. C296
;3294 ; PATTERN 4
;3295 ;$ 8000. 1555
;3296 ;$ 0000. 0000
;3297 ;$ 7000. 1555
;3298 ;$ 0000. 0000
;3299 ;$ 0040. 1FC3
;3300 ;$ 7612. EB1A
;3301 ; PATTERN 5
;3302 ;$ 5800. 1555
;3303 ;$ 0000. 0000
;3304 ;$ 5700. 1555
;3305 ;$ 0000. 0000
;3306 ;$ 0040. 1FFB
;3307 ;$ BF60. 1544
;3308 ; PATTERN 6
;3309 ;$ 5580. 1555
;3310 ;$ 0000. 0000
;3311 ;$ 5570. 1555
;3312 ;$ 0000. 0000
;3313 ;$ 0040. 1FFF
;3314 ;$ 43F6. 003C
;3315 ; PATTERN 7
;3316 ;$ 5558. 1555
;3317 ;$ 0000. 0000
;3318 ;$ 5557. 1555
;3319 ;$ 0000. 0000
;3320 ;$ 0022. 0FFF
;3321 ;$ BCD5. B002
;3322 ; PATTERN 8
;3323 ;$ 5555. 1555
;3324 ;$ 0000. B000
;3325 ;$ 5555. 1555
;3326 ;$ 0000. 7000
;3327 ;$ 001F. CFFF
```

```
;3328 ;$ BFCE, 6F01
;3329 ; PATTERN 9
;3330 ;$ 5555, 1555
;3331 ;$ 0000, 5800
;3332 ;$ 5555, 1555
;3333 ;$ 0000, 5700
;3334 ;$ 001F, CFFF
;3335 ;$ BFFD, 22F0
;3336 ; PATTERN 10
;3337 ;$ 5555, 1555
;3338 ;$ 0000, 5580
;3339 ;$ 5555, 1555
;3340 ;$ 0000, 5570
;3341 ;$ 001F, CFFF
;3342 ;$ C000, 0E2F
;3343 ; PATTERN 11
;3344 ;$ 5555, 1555
;3345 ;$ 0000, 5558
;3346 ;$ 5555, 1555
;3347 ;$ 0000, 5557
;3348 ;$ 001F, CFFF
;3349 ;$ C000, 3CE3
;3350 ; PATTERN 12
;3351 ;$ 5555, 1555
;3352 ;$ 8000, 5555
;3353 ;$ 5555, 1555
;3354 ;$ 7000, 5555
;3355 ;$ 001F, D000
;3356 ;$ 2000, 3FCE
;3357 ; PATTERN 13
;3358 ;$ 5555, 1555
;3359 ;$ 5800, 5555
;3360 ;$ 5555, 1555
;3361 ;$ 5700, 5555
;3362 ;$ 001F, D000
;3363 ;$ 0200, 3FFD
;3364 ; PATTERN 14
;3365 ;$ 5555, 1555
;3366 ;$ 5580, 5555
;3367 ;$ 5555, 1555
;3368 ;$ 5570, 5555
;3369 ;$ 001F, D000
;3370 ;$ 0020, 4001
;3371 ; PATTERN 15
;3372 ;$ 5555, 1555
;3373 ;$ 5558, 5555
;3374 ;$ 5555, 1555
;3375 ;$ 5557, 5555
;3376 ;$ 001F, D000
;3377 ;$ 0002, 4001
;3378
;3379
```

```
;3380  .PAGE "TEST 297 DIVIDE FLOAT FLOATING ONE BY FLOATING ONE"
;3381 ;*****************************************************************
;3382 ;**
;3383 ; DIVIDE FLOAT FLOATING ONE BY FLOATING ONE
;3384 ;
;3385 ; TEST DESCRIPTION
;3386 ;
;3387 ; THIS TEST DIVIDES A FLOATING (1) PATTERN IN THE
;3388 ; REMAINDER REGISTER BY A FLOATING (1) PATTERN
;3389 ; IN THE NORMALIZER REGISTER. THIS PATTERN AUGMENTS
;3390 ; THE THE FAULT DETECTION ACCOMPLISHED BY THE PREVIOUS
;3391 ; TESTS.
;3392 ;
;3393 ; TEST SEQUENCE   DESCRIPTION
;3394 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;3395 ; 2  PACK THE DATA
;3396 ; 3  PERFORM THE DIVIDE
;3397 ; 4  VERIFY RESULTS
;3398 ;
;3399 ; FPA UCODE USED
;3400 ;
;3401 ; LOCATION CONTENTS
;3402 ; 089: MSC/IR0,OPLD/INIT,BSC/R,SCR/CPU,EAC/LDA,WAIT,NAD/7F
;3403 ; 07F: BSC/R,SCR,CPU,FADC/AR,MSC/RR.0,NAD/0D6,WAIT
;3404 ; 0D6: BSC/R,SCR/CPU,FADC/AR0,NAD/0D7,WAIT
;3405 ; 0D7: FADC/A,BSC/FAL.HL,NRC/LD,NAD/0D8
;3406 ; 0D8: BSC/R,SCR/CPU,FADC/AR,NAD/0D9,WAIT
;3407 ; 0D9: BSC/R,SCR/CPU,FADC/AR0,NAD/0DA,WAIT
;3408 ; 0DA: FADC/A,BSC/FAL.HL,NRC/LD,LRR/LD.RR,BMXC/LB,NAD/0DB
;3409 ; 0DB: MCTL/CNT,BMXC/LB,NAD/0D0
;3410 ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;3411 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;3412 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;3413 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;3414 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;3415 ;
;3416 ; LOGIC DESCRIPTION
;3417 ;
;3418 ; THE NORMALIZER REGISTER,REMAINDER REGISTER,NALU, AND
;3419 ; DIVIDE CONTROLS ARE ON THE FNM MODULE. THE TMP/LSH
;3420 ; QUOTIENT REGISTERS ARE ON THE FML AND FMH MODULES.
;3421 ; THE CLEAR REMAINDER SIGNAL IS GENERATED ON THE FCT
;3422 ;
;3423 ; ERROR DESCRIPTION
;3424 ;
;3425 ; PACKED EXPECTED NORMALIZED QUOTIENT HI
;3426 ; PACKED RECEIVED NORMALIZED QUOTIENT HI
;3427 ; PACKED EXPECTED NORMALIZED QUOTIENT LO
;3428 ; PACKED RECEIVED NORMALIZED QUOTIENT LO
;3429 ; PACKED EXPECTED UNNORMALIZED QUOTIENT HI
;3430 ; PACKED RECEIVED UNNORMALIZED QUOTIENT HI
;3431 ; PACKED EXPECTED UNNORMALIZED QUOTIENT LO
;3432 ; PACKED RECEIVED UNNORMALIZED QUOTIENT LO
;3433 ; REMAINDER REGISTER HI
;3434 ; REMAINDER REGISTER LO
```

```
            ;3435 ; NORMALIZER REGISTER HI
            ;3436 ; NORMALIZER REGISTER LO
            ;3437 ; BIT COUNTER
            ;3438 ;
            ;3439 ; SYNC POINT
            ;3440 ;
            ;3441 ;--
            ;3442 ;
            ;3443 ;*****************************************************************
            ;3444 ;/////////////////////////////////////////////////////
            ;3445 ;*
            ;3446 ; FLOAT A1 THROUGH THE RR AND NR. DIVIDE AND CHECK RESULTS.
            ;3447 ;-
            ;3448 =0
U 10BA, 0018,0039,8980,09F8,0000,10D4  ;3449 DIVT7: NEWTST[.D]
U 10BB, 0818,0038,4D80,0800,0000,12E9  ;3450  D_K[.FF00]
U 12E9, 0819,0030,4180,0800,0000,12EA  ;3451  D_D.OR.K[.80]
U 12EA, 0001,0028,0180,0A88,0000,12EB  ;3452  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 12EB, 0000,003C,6180,0800,0084,72EC  ;3453  SC_K[.F]
U 12EC, 0000,003C,0580,0800,0084,B2ED  ;3454  SC_SC-K[.1]
U 12ED, 0801,0034,0180,0800,0000,12EE  ;3455  ALU_D.AND.MASK,D_ALU
U 12EE, 0001,0028,0180,0A90,0000,12EF  ;3456  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 12EF, 0818,0038,7980,0800,0000,12F0  ;3457  D_K[.30]
U 12F0, 0819,0030,8980,0800,0000,12F1  ;3458  D_D.OR.K[.D]
U 12F1, 0001,003C,0180,0990,0000,10BC  ;3459  RC[2]_D   ; BIT COUNT OF 61
U 10BC, 0000,003D,0180,0800,0000,10E6  ;3460 =0 ERLOOP
U 10BD, 0810,0038,0180,0910,0082,12F2  ;3461 DIV10: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 12F2, 0019,0034,7580,0800,0010,12F3  ;3462  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING RR
            ;3463       ; LO WITH PATTERN, RR HI
            ;3464       ; WITH ZERO.
U 12F3, 0000,013C,0180,0800,0000,10BE  ;3465  Z?
U 10BE, 0000,003C,0180,0800,0000,12F6  ;3466 =0 J/DIV11
U 10BF, 0803,001C,0180,0800,0000,12F4  ;3467  D_NOT.MASK  ; FLOATING 1
U 12F4, 0019,0024,5980,0AA8,0000,12F5  ;3468  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT BITS <6:0>
U 12F5, 0003,003C,0180,0AA0,0000,12FE  ;3469  R[4]_0,J/DIV12
U 12F6, 0003,003C,0180,0AA8,0000,12F7  ;3470 DIV11: R[5]_0   ; RR LO GETS 0
U 12F7, 0003,001C,0180,0AA0,0000,12F8  ;3471  R[4]_NOT.MASK  ; FLOATING 1
U 12F8, 0800,003C,0180,0A20,0000,12F9  ;3472  D_R[4]
U 12F9, 0001,003C,0180,09B0,0000,12FA  ;3473  RC[6]_D   ; SAVE RR HI
U 12FA, 0001,003C,0180,09A0,0000,12FB  ;3474  RC[4]_D   ; SAVE NR HI
U 12FB, 0800,003C,0180,0A28,0000,12FC  ;3475  D_R[5]
U 12FC, 0001,003C,0180,09A8,0000,12FD  ;3476  RC[5]_D   ; SAVE RR LO
U 12FD, 0001,003C,0180,0998,0000,12FE  ;3477  RC[3]_D   ; SAVE NR LO
U 12FE, 0003,003C,0180,09F0,0000,12FF  ;3478 DIV12: RC[0E]_0  ; EXPECTED NORMALIZED QUOTIENT HI
U 12FF, 0003,003C,0180,09E0,0000,1300  ;3479  RC[0C]_0  ; EXPECTED  NORMALIZED QUOTIENT LO
U 1300, 0003,003C,0180,09D0,0000,1301  ;3480  RC[0A]_0  ; EXPECTED UNNORMALIZED QUOTIENT HI
U 1301, 0003,003C,0180,09C0,0000,10C0  ;3481  RC[8]_0   ; EXPECTED UNNORMALIZED QUOTIENT LO
            ;3482 ; PACK OPERANDS FOR DIVIDE OPERATION
U 10C0, 0000,003D,0180,0800,0000,1131  ;3483 =0 CALL[UNSEQA]
U 10C1, 0800,003C,0180,0A30,0000,1302  ;3484  D_R[6]
U 1302, 080D,0034,0180,0A08,0000,1303  ;3485  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 1303, 0001,003C,0180,0AB0,0000,1304  ;3486  R[6]_D   ; HOLD RR HI MASKED FOR DIVIDE
U 1304, 0001,003C,0180,0AC0,0000,1305  ;3487  R[8]_D   ; HOLD NR HI MASKED FOR DIVIDE
U 1305, 0800,003C,0180,0A38,0000,1306  ;3488  D_R[7]   ;
U 1306, 0001,003C,0180,0AC8,0000,10C2  ;3489  R[9]_D   ; HOLD NR LO FOR DIVIDE
```

```
U 10C2, 2018,0039,8580,0800,4200,10FA  ;3490 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; DIVD R,R
U 10C3, 0000,003C,0180,0800,0000,10C4  ;3491  NOP
U 10C4, 0000,003D,0180,0800,0000,1010  ;3492 =0 CALL[LD.DIV]  ; LOAD OPERANDS, DIVIDE
        ;3493      ; READ DATA INTO Q AND D.
U 10C5, 0001,203C,0180,09D8,0000,1307  ;3494  RC[0B]_Q  ; SAVE NORMALIZED QUOT. LO
U 1307, 081C,2034,0180,0A08,0000,1308  ;3495  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 1308, 0001,003C,0180,09E8,0000,1309  ;3496  RC[0D]_D  ; SAVE NORMALIZED QUOT. HI
U 1309, 0000,003C,0180,0908,0000,130A  ;3497  LC_RC[1]
U 130A, 0810,0038,0180,0800,0000,130B  ;3498  D_LC
U 130B, 080D,0034,0180,0A10,0000,130C  ;3499  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 130C, 0001,003C,0180,09C8,0000,130D  ;3500  RC[9]_D  ; SAVE UNNORMALIZED QUOT. HI
U 130D, 0000,003C,0180,0900,0000,130E  ;3501  LC_RC[0]
U 130E, 0010,0038,0180,09B8,0000,130F  ;3502  RC[7]_LC  ; SAVE UNNORMALIZED QUOT. LO
U 130F, 0011,2020,0180,0960,0010,1310  ;3503  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED QUOTIENT LO
U 1310, 0000,013C,0180,0800,0000,10C6  ;3504  Z?
U 10C6, 0000,003D,0180,0800,0000,10E7  ;3505 =0 ERROR1
U 10C7, 0010,0038,01C0,0968,0000,1311  ;3506  Q_RC[0D]
U 1311, 0011,2020,0180,0970,0010,1312  ;3507  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED QUOTIENT HI
U 1312, 0000,013C,0180,0800,0000,10C8  ;3508  Z?
U 10C8, 0000,003D,0180,0800,0000,10E7  ;3509 =0 ERROR1
U 10C9, 0010,0038,01C0,0940,0000,1313  ;3510  Q_RC[8]
U 1313, 0011,2020,0180,0938,0010,1314  ;3511  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT LO
U 1314, 0000,013C,0180,0800,0000,10CA  ;3512  Z?
U 10CA, 0000,003D,0180,0800,0000,10E7  ;3513 =0 ERROR1
U 10CB, 0010,0038,01C0,0950,0000,1315  ;3514  Q_RC[0A]
U 1315, 0011,2020,0180,0948,0010,1316  ;3515  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED QUOTIENT HI
U 1316, 0000,013C,0180,0800,0000,10CC  ;3516  Z?
U 10CC, 0000,003D,0180,0800,0000,10E7  ;3517 =0 ERROR1
U 10CD, 0810,0038,0180,0910,0000,1317  ;3518  D_RC[2]
U 1317, 0018,0038,2DC0,0800,0000,1318  ;3519  Q_K[.28]
U 1318, 0019,2000,05C0,0800,0000,1319  ;3520  Q_Q-K[.1]
U 1319, 001D,0000,0180,0800,0010,131A  ;3521  ALU_D-Q,CLK.UBCC  ; CHECK BIT COUNTER
U 131A, 0000,013C,0180,0800,0000,10CE  ;3522  Z?    ; 25 BITS PROCESSED?
U 10CE, 0019,0000,0580,0990,0000,10BD  ;3523 =0 ALU_D-K[.1],RC[2]_ALU,J/DIV10 ; DECREMENT BIT COUNTER
U 10CF, 0000,003C,0180,0800,0000,10D0  ;3524  NOP
        ;3525
```

```
            ;3526  .PAGE "TEST 298 RESERVED"
            ;3527 =0
U 10D0. 0000.003D.0180.0800.0000.10D4  ;3528 TSTXX2: NEWTST
U 10D1. 0000.003C.0180.0800.0000.131B  ;3529  NOP
            ;3530
```

```
            ;3531  .PAGE "DUMMY NEWTST"
U 131B. 0000,003D,0180,0800,0000,10D4  ;3532 TSTXX3: NEWTST
            ;3533
```

ZZ-ESKAR-V22 Line Number Index
ESKAR64.MCR MICRO2 1P(00) 26-JUL-85 17:09:33 SEQ 2781
: Location / Line Number Index Page 8

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000 1907: 1874 1881= 1897= 1898= 1899= 2066= 2067=
U 1008 1918= 1919= 1875 1999= 2010= 2011= 1876 2000=
U 1010 2049= 2050= 2090= 2091= 2167= 2168= 2170= 2171=
U 1018 2172= 2173= 1877 2098= 2175= 2176= 1878 2099=
U 1020 2190= 2191= 2274= 2275= 2276= 2277= 2281= 2282=
U 1028 2286= 2287= 1879 2104= 2288= 2289= 1880 2105=
U 1030 2321= 2322= 2328= 2329= 2334= 2335= 1900 2107=
U 1038 2339= 2340= 2341= 2342= 2390= 2391= 1901 2108=
U 1040 2462= 2463= 2473= 2474= 2479= 2480= 1902 2676=
U 1048 2497= 2498= 2516= 2517= 2528= 2529= 1939: 2677=
U 1050 2536= 2537= 2538= 2540= 2551= 2552= 2555= 2556=
U 1058 2559= 2560= 1942: 1903 2563= 2564= 1945: 1904
U 1060 2566= 2567= 2641= 2642= 2653= 2654= 2659= 2660=
U 1068 2668= 2669= 2686= 2687= 2693= 2694= 2511= 2512=
U 1070 2713= 2714= 2725= 2726= 2735= 2736= 2737= 2739=
U 1078 2750= 2751= 2754= 2755= 2758= 2759= 2708= 2709=
U 1080 2762= 2763= 2765= 2766= 2836= 2837= 2847= 2848=
U 1088 2853= 2854= 2870= 2871= 2877= 2878= 2879= 2881=
U 1090 2892= 2893= 2896= 2897= 2900= 2901= 2904= 2905=
U 1098 2907= 2908= 2981= 2982= 2995= 2997= 3017= 3018=
U 10A0 3020= 3022= 3038= 3039= 3045= 3046= 3049= 3050=
U 10A8 3178= 3179= 3192= 3194= 3203= 3204= 3219= 3220=
U 10B0 3229= 3230= 3232= 3234= 3250= 3251= 3257= 3258=
U 10B8 3261= 3262= 3449= 3450= 3460= 3461= 3466= 3467=
U 10C0 3483= 3484= 3490= 3491= 3492= 3494= 3505= 3506=
U 10C8 3509= 3510= 3513= 3514= 3517= 3518= 3523= 3524=
U 10D0 3528= 3529= 1905 1906 1916 1917 1920 1921
U 10D8 1922 1923 1924 1925 1926 1927 1928 1929
U 10E0 1930 1931 1932 1933 1934 1935 1936 1937
U 10E8 1938 1940 1941 1973 1974 1980 1981 1982
U 10F0 1995 1996 1997 1998 2008 2009 2012 2019
U 10F8 2020 2021 2031 2032 2033 2034 2035 2036
U 1100 1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108 1869: 2037 2051 2052 1870: 1871: 1872: 1873:
U 1110 2053 2054 2055 2056 2057 2058 2059 2060
U 1118 2061 2062 2064 2065 2072 2073 2074 2075
U 1120 2077 2078 2092 2093 2094 2095 2096 2097
U 1128 2100 2101 2102 2103 2106 2109 2115 2116
U 1130 2117 2137 2138 2139 2140 2141 2142 2143
U 1138 2144 2145 2146 2147 2148 2149 2150 2151
U 1140 2152 2165 2166 2169 2174 2177 2192 2193
U 1148 2194 2201 2202 2203 2204 2205 2206 2207
U 1150 2208 2209 2210 2278 2279 1965: 2280 2283
U 1158 2284 2285 2290 2291 2292 2293 2294 2295
U 1160 2296 2297 2298 2299 2302 2303 2304 2305
U 1168 2306 2307 2308 2309 2310 2311 2312 2313
U 1170 2314 2315 2316 2317 2318 2319 2320 2330
U 1178 2331 2332 2336 2337 2338 2343 2344 2345
U 1180 2346 2347 2348 2349 2350 2351 2352 2355
U 1188 2356 2357 2358 2359 2360 2361 2362 2363
U 1190 2364 2365 2366 2367 2368 2369 2370 2371
U 1198 2372 2373 2374 2375 2376 2377 2378 2379

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2380 2381 2382 2383 2384 2385 2386 2387
U 11A8  2388 2389 2464 2465 2466 2467 2468 2469
U 11B0  2470 2471 2472 2475 2478 2481 2482 2483
U 11B8  2484 2486 2487 2488 2489 2490 2491 2492
U 11C0  2493 2494 2495 2499 2500 2501 2502 2503
U 11C8  2505 2506 2507 2508 2509 2510 2513 2518
U 11D0  2519 2520 2521 2522 2524 2525 2526 2527
U 11D8  2530 2531 2532 2533 2534 2535 2541 2542
U 11E0  2543 2544 2545 2546 2547 2548 2549 2550
U 11E8  2553 2554 2557 2558 2561 2562 2565 2643
U 11F0  2644 2645 2646 2647 2648 2649 2650 2651
U 11F8  2652 2655 2658 2661 2662 2663 2664 2665
U 1200  2666 2667 2670 2671 2672 2673 2674 2675
U 1208  2680 2681 2682 2683 2684 2685 2688 2689
U 1210  2690 2695 2696 2697 2698 2699 2701 2702
U 1218  2703 2704 2705 2706 2707 2710 2715 2716
U 1220  2717 2718 2719 2721 2722 2723 2724 2727
U 1228  2728 2729 2730 2731 2732 2733 2734 2740
U 1230  2741 2742 2743 2744 2745 2746 2747 2748
U 1238  2749 2752 2753 2756 2757 2760 2761 2764
U 1240  2838 2839 2840 2841 2842 2843 2844 2845
U 1248  2846 2849 2852 2855 2856 2857 2858 2859
U 1250  2860 2861 2862 2863 2864 2865 2866 2867
U 1258  2868 2872 2873 2874 2875 2876 2882 2883
U 1260  2884 2885 2886 2887 2888 2889 2890 2891
U 1268  2894 2895 2898 2899 2902 2903 2906 2983
U 1270  2984 2985 2986 2987 2988 2989 2990 2991
U 1278  2992 2993 2994 2998 2999 3000 3001 3002
U 1280  3003 3004 3005 3006 3007 3008 3009 3010
U 1288  3011 3012 3013 3014 3015 3016 3023 3024
U 1290  3025 3026 3027 3028 3029 3030 3031 3032
U 1298  3033 3034 3036 3037 3040 3041 3042 3043
U 12A0  3044 3047 3048 3051 3052 3053 3180 3181
U 12A8  3182 3183 1966: 3184 3185 3186 3187 3188
U 12B0  3189 3190 3191 3195 3196 3197 3198 3199
U 12B8  3200 3201 3202 3205 3206 3207 3208 3209
U 12C0  3210 3211 3212 3213 3214 3215 3216 3217
U 12C8  3218 3221 3222 3223 3224 3225 3226 3227
U 12D0  3228 3235 3236 3237 3238 3239 3240 3241
U 12D8  3242 3243 3244 3245 3246 3248 3249 3252
U 12E0  3253 3254 3255 3256 3259 3260 3263 3264
U 12E8  3265 3451 3452 3453 3454 3455 3456 3457
U 12F0  3458 3459 3462 3465 3468 3469 3470 3471
U 12F8  3472 3473 3474 3475 3476 3477 3478 3479
U 1300  3480 3481 3485 3486 3487 3488 3489 3495
U 1308  3496 3497 3498 3499 3500 3501 3502 3503
U 1310  3504 3507 3508 3511 3512 3515 3516 3519
U 1318  3520 3521 3522 3532
U 1320 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

	Words not in bounds
ESK00DEF	0
ESK00MAC	0
ESKARMAC	0
ESKAR64	812
Used	812
Remaining	

Total microwords used in memory U: 812
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2784
ESKAR64.MCR                    MICRO2  1P(00)    26-JUL-85  17:09:33     Page   9


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR64.MIC


Pass 1 warnings:     0 Pass 2 warnings:     0
 Unsplit Binary Lines:     0
Pass 1 errors:       0 Pass 2 errors:       0
```

ZZ-ESKAR-V22 Table of Contents SEQ 2785
ESKAR65.MCR MICRO2 1P(00) 26-JUL-85 17

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2331 TEST 299 INTEGER MULTIPLY (PRODUCT LO)
; 2488 TEST 29A INTEGER MULTIPLY (PRODUCT HI)
; 2601 TEST 29B MULTIPLY FLOAT, FLOATING ONE BY 0.1
; 2800 TEST 29C MULTIPLY FLOAT, 0.1 BY FLOATING ONE
; 3009 TEST 29D EMODF, FLOATING ONE BY 0.1
; 3152 TEST 29E EMODF, FLOATING ONE BY 0.1
; 3305 TEST 29F RESERVED
; 3310 TEST 2A0 RESERVED
; 3315 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR65
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844  ;*
            ;1845  ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846  ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847  ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848  ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849  ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850  ;
            ;1851  ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852  ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853  ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854  ; R[0F] AND DO A RETURN0.
            ;1855  ;
            ;1856  ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857  ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858  ; TO THE CONSOLE.
            ;1859  ;
            ;1860  ;-
U 1100. 0000.003C.1980.0800.0084.7003   ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101. 0000.003C.0580.0800.0084.7003   ;1862 1101: SC_K[.1],J/TRPH1  ; DATA UNALIGNED
U 1102. 0000.003C.0980.0800.0084.7003   ;1863 1102: SC_K[.2],J/TRPH1  ; X PAGE ERROR
U 1103. 0000.003C.0D80.0800.0084.7003   ;1864 1103: SC_K[.3],J/TRPH1  ; TB MODIFY
U 1104. 0000.003C.1180.0800.0084.7003   ;1865 1104: SC_K[.4],J/TRPH1  ; TB PROT ERROR
U 1105. 0000.003C.D580.0800.0084.7003   ;1866 1105: SC_K[.6],J/TRPH1  ; TB MISS
U 1106. 0000.003C.D980.0800.0084.7003   ;1867 1106: SC_K[.9],J/TRPH1  ; RES FLOAT OPER
U 1107. 0000.003C.5D80.0800.0084.7003   ;1868 1107: SC_K[.7],J/TRPH1  ; TB PARITY
U 1108. 0000.003C.0180.0800.0084.7003   ;1869 1108: SC_K[.8],J/TRPH1  ; CACHE PARITY
U 110C. 0000.003C.8580.0800.0084.7003   ;1870 110C: SC_K[.C],J/TRPH1  ; RDS
U 110D. 0000.003C.8980.0800.0084.7003   ;1871 110D: SC_K[.D],J/TRPH1  ; TIME OUT
U 110E. 0000.003C.6580.0800.0084.7003   ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F. 0000.003C.6180.0800.0084.7003   ;1873 110F: SC_K[.F],J/TRPH1  ; CS PARITY
U 1001. 0000.003C.81F0.2C00.0000.100A   ;1874 TRPH0: Q_ID[USTACK]
U 100A. 0C00.003C.01E0.0800.0000.100E   ;1875  Q_D,D_Q
U 100E. 0000.003C.8180.3C00.0000.104F   ;1876  D[USTACK]_D
U 104F. 0C00.003C.01E0.0800.0000.105B   ;1877  Q_D,D_Q
U 105B. 0000.003C.01C0.0A78.0000.105F   ;1878  Q_R[0F]
U 105F. 0001.2024.0180.0800.0010.10CB   ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 10CB. 0000.013C.0180.0800.0000.1002   ;1880  Z?     ; BRANCH IF NO
U 1002. 0001.2036.0180.0AF8.0000.0000   ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882  ;*
            ;1883  ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884  ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885  ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886  ;
            ;1887  ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888  ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889  ;       FAULT STATUS REGISTER
            ;1890  ;       SBI ERROR REGISTER
            ;1891  ;       TIMEOUT ADDRESS REGISTER
            ;1892  ;       MAINTENANCE REGISTER
            ;1893  ;       CACHE PARITY REGISTER
            ;1894  ;       TB ERROR REGISTER 1
            ;1895  ;       TB ERROR REGISTER 0
```

```
            ;1896 ;-
U 1003. 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004. 0000.003D.D980.0800.0084.70EC  ;1898 =0 SETRCF[.9]
U 1005. 0000.003C.49F0.2C00.0000.10CC  ;1899  Q_ID[TBER0]
U 10CC. 0001.203C.4DF0.2DB0.0000.10CD  ;1900  RC[6]_Q,Q_ID[TBER1]
U 10CD. 0001.203C.65F0.2DB8.0000.10CE  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 10CE. 0001.203C.6DF0.2DD8.0000.10CF  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 10CF. 0001.203C.75F0.2DE0.0000.10D0  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 10D0. 0001.203C.79F0.2DC8.0000.10D1  ;1904  RC[9]_Q,Q_ID[PARITY]
U 10D1. 0001.203C.69F0.2DC0.0000.10D2  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 10D2. 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000. 0001.203D.0180.09F0.0000.10E6  ;1907 1000: RC[0E]_Q,ERROR1
            ;1908 ;*
            ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
            ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
            ;1911 ;  NEWTST
            ;1912 ;  ERLOOP
            ;1913 ;  ERROR1
            ;1914 ;  ERROR2
            ;1915 ;
U 10D3. 0000.003C.8580.0800.0084.70D4  ;1916 SCOPE: SC_K[.C]
U 10D4. 0803.001C.0180.0800.0000.1008  ;1917  ALU_NOT.MASK,D_ALU
U 1008. 0000.003D.7580.3C00.0000.1160  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1009. 0000.003C.65F8.0800.0084.70D5  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 10D5. 0818.0038.8D80.0800.0000.10D6  ;1920  D_K[.1F]    ; ...
U 10D6. 0D00.003C.0180.0800.0000.10D7  ;1921  D_DAL.SC
U 10D7. 0000.003C.3D80.3C00.0000.10D8  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 10D8. 0818.0038.0580.0800.0000.10D9  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 10D9. 0D00.003C.0180.0800.0000.10DA  ;1924  D_DAL.SC
U 10DA. 0F00.003C.3180.3C00.0000.10DB  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10DB. 0000.003C.0180.0800.0000.10DC  ;1926  NOP
U 10DC. 0000.003C.3980.3C00.0000.10DD  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 10DD. 0000.003C.2980.3C00.0000.10DE  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 10DE. 0000.003C.4980.3C00.0000.10DF  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 10DF. 0818.0038.C180.0800.0000.10E0  ;1930  D_K[.FFFF]
U 10E0. 0000.003C.6580.3C00.0000.10E1  ;1931  ID[SBI.ERR]_D
U 10E1. 0F00.003C.6D80.3C00.0000.10E2  ;1932  ID[FAULT]_D,D_0
U 10E2. 0000.003C.6D80.3C00.0000.10E3  ;1933  ID[FAULT]_D
U 10E3. 0000.003C.6580.3C00.0000.10E4  ;1934  ID[SBI.ERR]_D
U 10E4. 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON   ; CALL THE CONSOLE
U 10E5. 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON   ; ERRLOOP CALL
U 10E6. 0810.0038.0180.0978.0000.10E7  ;1937 ERROR1: D_RC[0F]
U 10E7. 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E. 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 10E8. 0810.0038.0180.0978.0000.10E9  ;1940 ERROR2: D_RC[0F]
U 10E9. 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A. 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
            ;1943 105E:
            ;1944 CALLCON:
U 105E. 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON   ; CALL THE CONSOLE
            ;1946 ;*
            ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
            ;1948 ;-
U 13F0. 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1. 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2. 0000.003C.0180.0800.0000.13F3  ;1951 13F2: NOP
U 13F3. 0000.003C.0180.0800.0000.13F4  ;1952 13F3: NOP
U 13F4. 0000.003C.0180.0800.0000.13F5  ;1953 13F4: NOP
U 13F5. 0000.003C.0180.0800.0000.13F6  ;1954 13F5: NOP
U 13F6. 0000.003C.0180.0800.0000.13F7  ;1955 13F6: NOP
U 13F7. 0000.003C.0180.0800.0000.13F8  ;1956 13F7: NOP
U 13F8. 0000.003C.0180.0800.0000.13F9  ;1957 13F8: NOP
U 13F9. 0000.003C.0180.0800.0000.13FA  ;1958 13F9: NOP
U 13FA. 0000.003C.0180.0800.0000.13FB  ;1959 13FA: NOP
U 13FB. 0000.003C.0180.0800.0000.13FC  ;1960 13FB: NOP
U 13FC. 0000.003C.0180.0800.0000.13FD  ;1961 13FC: NOP
U 13FD. 0000.003C.0180.0800.0000.13FE  ;1962 13FD: NOP
U 13FE. 0000.003C.0180.0800.0000.13FF  ;1963 13FE: NOP
U 13FF. 0000.003C.0180.0800.0000.1155  ;1964 13FF: NOP
U 1155. 0001.203E.59F0.2DE8.0000.0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA. 0001.203E.59F0.2DE8.0000.0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 10EA. 0810.0038.4D80.0978.0000.10EB  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 10EB. 0019.4016.4D80.09F8.0000.0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 10EC. 0010.0038.01C0.0978.0000.10ED  ;1980 SETRCF: Q_RC[0F]
U 10ED. 0019.2034.4DC0.0800.0000.10EE  ;1981  Q_Q.AND.K[.FF00]
U 10EE. 0019.2032.1D80.09F8.0000.0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983
            ;1984
            ;1985
            ;1986
            ;1987
            ;1988
            ;1989
            ;1990
            ;1991 ;******************************************************
            ;1992 ;
            ;1993 ; MAIN MACHINE/FPA SUBROUTINES
            ;1994 ;
            ;1995 ;******************************************************
            ;1996
            ;1997 ; SHIFT D REGISTTER CONTENTS
            ;1998
            ;1999
            ;2000 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
            ;2001
            ;2002
U 10EF. 0000.003C.01F8.0800.0000.10F0  ;2003 SHIFTD: Q_0
U 10F0. 0500.003C.0180.0800.0000.10F1  ;2004  D_D.LEFT  ;
U 10F1. 0000.003C.0580.0800.1414.B0F2  ;2005  EALU_STATE-K[.1],CLK.UBCC ;
```

```
U 10F2, 0000,123C,0180,0800,0000,100B  ;2006  EALU.Z?
U 100B, 0000,003C,0180,0800,0000,10EF  ;2007 =1011 J/SHIFTD  :
U 100F, 0000,003E,0180,0800,0000,0001  ;2008  RETURN1
            ;2009
            ;2010 ;*
            ;2011 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
            ;2012 ; OR'S IN BIT 15, SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP
            ;2013 ; TO THAT ADDRESS.
            ;2014 ;-
            ;2015
U 10F3, 0819,0030,4580,0800,0000,10F4  ;2016 TRAP1: D_D.OR.K[.8000]   ; SET UP TRAP ENABLE BIT
U 10F4, 0000,003C,6580,0800,1404,700C  ;2017  STATE_K[.10]   ; SHIFT TO BIT POSITIONS <23:16>
U 100C, 0000,003D,0180,0800,0000,10EF  ;2018 =0 CALL[SHIFTD]
U 100D, 0000,003C,5980,3C00,0000,10F5  ;2019  ID[ACC.2]_D   ; LOAD FPA STATUS REGISTER (ID 16)
U 10F5, 0000,00FE,0380,0800,0000,0001  ;2020  TRAP.FPA,RETURN1
            ;2021
            ;2022 ;*
            ;2023 ; THE FOLOWING ROUTINES GENERATE FPA UPC ADDRESSES, AND
            ;2024 ; JUMP TO THE TRAP ROUTINE.
            ;2025 ;-
            ;2026
U 10F6, 0818,0038,A580,0800,0000,10F7  ;2027 TRP.68: D_K[.60]
U 10F7, 0819,0030,0180,0800,0000,10F3  ;2028  D_D.OR.K[.8],J/TRAP1
            ;2029
            ;2030
U 10F8, 0818,0038,A580,0800,0000,10F9  ;2031 TRP.6A: D_K[.60]
U 10F9, 0819,0030,F580,0800,0000,10F3  ;2032  D_D.OR.K[.A],J/TRAP1
            ;2033
U 10FA, 0818,0038,D180,0800,0000,10FB  ;2034 TRP.C8: D_K[.C0]
U 10FB, 0819,0030,0180,0800,0000,10F3  ;2035  D_D.OR.K[.8],J/TRAP1
            ;2036
U 10FC, 0818,0038,D180,0800,0000,10FD  ;2037 TRP.CC: D_K[.C0]
U 10FD, 0819,0030,8580,0800,0000,10F3  ;2038  D_D.OR.K[.C],J/TRAP1
            ;2039
U 10FE, 0818,0038,5D80,0800,0000,10FF  ;2040 TRP.CE: D_K[.7]
U 10FF, 0500,003C,0180,0800,0000,1109  ;2041  D_D.LEFT
U 1109, 0819,0030,D180,0800,0000,10F3  ;2042  D_D.OR.K[.C0],J/TRAP1
            ;2043
U 110A, 0818,0038,8980,0800,0000,110B  ;2044 D0: D_K[.D]
U 110B, 0100,003C,0180,0800,0000,1110  ;2045  D_D.LEFT2
U 1110, 0100,003E,0180,0800,0000,0001  ;2046  D_D.LEFT2,RETURN1
            ;2047
            ;2048 =0
U 1010, 0000,003D,0180,0800,0000,110A  ;2049 TRP.D1: CALL[D0]
U 1011, 0819,0030,0580,0800,0000,10F3  ;2050  D_D.OR.K[.1],J/TRAP1
            ;2051
U 1111, 0818,0038,CD80,0800,0000,1112  ;2052 E0: D_K[.F0]
U 1112, 0819,0026,6580,0800,0000,0001  ;2053  D_D[ANDNOT]K[.10],RETURN1
            ;2054
            ;2055 =0
U 1012, 0000,003D,0180,0800,0000,1111  ;2056 TRP.E1: CALL[E0]
U 1013, 0819,0030,0580,0800,0000,10F3  ;2057  D_D.OR.K[.1],J/TRAP1
            ;2058
            ;2059 =0
U 1014, 0000,003D,0180,0800,0000,1111  ;2060 TRP.E3: CALL[E0]
```

```
U 1015. 0819,0030,0D80,0800,0000,10F3  ;2061  D_D.OR.K[.3],J/TRAP1
            ;2062
            ;2063 =0
U 1018. 0000,003D,0180,0800,0000,1111  ;2064 TRP.E2: CALL[E0]
U 1019. 0819,0030,0980,0800,0000,10F3  ;2065  D_D.OR.K[.2],J/TRAP1
            ;2066
            ;2067 =0
U 101A. 0000,003D,0180,0800,0000,1111  ;2068 TRP.E4: CALL[E0]
U 101B. 0819,0030,1180,0800,0000,10F3  ;2069  D_D.OR.K[.4],J/TRAP1
            ;2070
            ;2071 =0
U 101C. 0000,003D,0180,0800,0000,1111  ;2072 TRP.EA: CALL[E0]
U 101D. 0819,0030,F580,0800,0000,10F3  ;2073  D_D.OR.K[.A],J/TRAP1
            ;2074
            ;2075
U 1113. 0818,0038,F980,0800,0000,1114  ;2076 TRP.EE: D_K[.7E]
U 1114. 0819,0030,4180,0800,0000,1115  ;2077  D_D.OR.K[.80]
U 1115. 0819,0024,6580,0800,0000,10F3  ;2078  D_D.ANDNOT.K[.10],J/TRAP1
            ;2079
            ;2080
            ;2081
U 1116. 0818,0038,CD80,0800,0000,1117  ;2082 TRP.FB: D_K[.F0]
U 1117. 0819,0030,F580,0800,0000,1118  ;2083  D_D.OR.K[.A]
U 1118. 0819,0014,0580,0800,0000,10F3  ;2084  D_D+K[.1],J/TRAP1
            ;2085
            ;2086 ;+
            ;2087 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2088 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT
            ;2089 ; THE IB IS COMPLETELY FULL OF DATA.
            ;2090 ;
            ;2091 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2092 ;-
U 1119. 3000,003C,0180,6000,0000,111A  ;2093 IBLOAD: LOAD.IB,IBC/START
U 111A. 0000,003C,0180,F800,0000,111B  ;2094 SYNC02: MCT/ALLOW.IB.READ
U 111B. 0000,003C,0180,F800,0000,111C  ;2095  MCT/ALLOW.IB.READ
U 111C. 0000,003C,0180,F800,0000,111D  ;2096  MCT/ALLOW.IB.READ
U 111D. 0000,003C,0180,F800,0000,111E  ;2097  MCT/ALLOW.IB.READ
U 111E. 0000,003C,0180,F800,0000,111F  ;2098  MCT/ALLOW.IB.READ
U 111F. 1000,003E,0180,F800,0000,0001  ;2099  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2100
            ;2101 ;+
            ;2102 ; THIS ROUTINE LOADS THE FAD AR AND BR LATCHES WITH MULTIPLIER AND
            ;2103 ; MULTIPLICAND OPERANDS. R6,R7,R8,AND R9 MUST BE LOADED IN THE
            ;2104 ; CALLING PROGRAM. THE LA EXPONENT LATCH IS LOADED SO THAT WHEN THE
            ;2105 ; NORMALIZED FRACTION PRODUCTS ARE READ BACK ON BUSA, THE NORMALIZER
            ;2106 ; MUX WILL PASS THE NORMALIZED PRODUCT, AND NOT FORCE ZEROS ON THE BUS.
            ;2107 ;-
            ;2108
            ;2109 =0
U 1020. 0000,003D,0180,0800,0000,10F6  ;2110 LD.MUL: CALL[TRP.68]
U 1021. 0018,0038,4180,0AF0,0000,1120  ;2111  R[0E]_K[.80]
U 1120. 0000,003C,0180,0A70,0000,1121  ;2112  LAB_R[0E]  ; FPA AT 068.
U 1121. 0000,007C,0180,0A70,0000,1122  ;2113  LAB_R[0E],CPSYNC ; LOAD LA
U 1122. 0000,007C,0180,0800,0000,1123  ;2114  CPSYNC    ; FPA AT 069.
U 1123. 0000,003C,0180,0A30,0000,1124  ;2115  LAB_R[6]  ; FPA AT 06A.
```

```
U 1124. 0000.007C.0180.0A30.0000.1125  ;2116  LAB_R[6],CPSYNC  ; LOAD AR
U 1125. 0000.003C.0180.0A38.0000.1126  ;2117  LAB_R[7]  ; FPA AT 06C.
U 1126. 0000.007C.0180.0A38.0000.1127  ;2118  LAB_R[7],CPSYNC  ; LOAD AR0
U 1127. 0000.003C.0180.0A40.0000.1128  ;2119  LAB_R[8]  ; FPA AT 06D.
U 1128. 0000.007C.0180.0A40.0000.1129  ;2120  LAB_R[8],CPSYNC  ; LOAD BR
U 1129. 0000.003C.0180.0A48.0000.112A  ;2121  LAB_R[9]  ; FPA AT 083.
U 112A. 0000.007C.0180.0A48.0000.112B  ;2122  LAB_R[9],CPSYNC  ; LOAD BR0
U 112B. 0000.003E.0180.0800.0000.0001  ;2123  RETURN1   ; FPA AT 028. LOAD COMPLETE.
            ;2124
            ;2125 ;*
            ;2126 ; THIS ROUTINE LOADS THE IB WITH A MUL FLOAT INSTRUCTION THEN
            ;2127 ; TAKES MULTIPLIER AND MULTIPLICAND OPERANDS PREVIOUSLY LOADED
            ;2128 ; IN THE FAD AR AND BR LATCHES, AND LOADS THE MULTIPLIER OPERAND
            ;2129 ; REGISTERS FOR A MUL FLOAT. THE ROUTINE THEN JUMPS TO SUBROUTINE
            ;2130 ; PROD TO READ BACK THE PRODUCT.
            ;2131
            ;2132 =0
U 1022. 2018.0039.0180.0800.4200.1119  ;2133 MULF: ALU_K[.8],FLUSH.IB,CALL[IBLOAD] ;
U 1023. 0000.003C.0180.0800.0000.1024  ;2134  NOP
U 1024. 0000.003D.0180.0800.0000.1012  ;2135 =0 CALL[TRP.E1]
U 1025. 0000.003C.0180.0800.0000.112C  ;2136  NOP    ; FPA AT 0E1. MC1_AR1
U 112C. 0000.003C.0180.0800.0000.112D  ;2137  NOP    ; FPA AT 0E2. MP1_BR1
U 112D. 0000.003C.0180.0800.0000.112E  ;2138  NOP     ; FPA AT 0C8. START MULTIPLY
U 112E. 0000.003C.0180.0800.0000.112F  ;2139  NOP    ; FPA AT 0C9. CONTINUE MULTIPLY
U 112F. 0000.003C.0180.0800.0000.1130  ;2140  NOP    ; FPA AT 028. MULF R,R DONE.
U 1130. 0000.003C.0180.0800.0000.1131  ;2141  NOP    ; ALLOW PRODUCT TO FINALIZE
U 1131. 0000.003C.0180.0800.0000.1040  ;2142  NOP,J/PROD  ;
            ;2143
            ;2144 ;*
            ;2145 ; THE MULL ROUTINE LOADS THE IB WITH A MULTIPLY DOUBLE OPCODE
            ;2146 ; THEN TRAPS TO THE UCODE THAT LOADS THE MULTIPLIER OPERAND
            ;2147 ; REGISTERS. IT THEN LOADS THE IB WITH THE MUL LONG OPCODE,
            ;2148 ; AND JUMPS TO THE ROUTINE WHICH INITIATES THE MULTIPLY.
            ;2149 ;-
            ;2150
            ;2151 =0
U 1028. 2018.0039.6580.0800.4200.1119  ;2152 MULL: ALU_K[.10],FLUSH.IB,CALL[IBLOAD] ; ISSUE DOUBLE OPCODE
            ;2153        ; TO ENABLE FAD MUX TO
            ;2154        ; PASS AR0,BR0.
U 1029. 0000.003C.0180.0800.0000.102A  ;2155  NOP
U 102A. 0000.003D.0180.0800.0000.101A  ;2156 =0 CALL[TRP.E4]
U 102B. 0000.003C.0180.0800.0000.102C  ;2157  NOP    ; FPA AT 0E4. MC1_BR0
U 102C. 0000.003D.0180.0800.0000.1113  ;2158 =0 CALL[TRP.EE]
U 102D. 0000.003C.0180.0800.0000.1030  ;2159  NOP    ; FPA AT 0EE. MP0_AR0
U 1030. 2018.0039.8580.0800.4200.1119  ;2160 =0 ALU_K[.C],FLUSH.IB,CALL[IBLOAD] ; MULL R,R
U 1031. 0000.003C.0180.0800.0000.1032  ;2161  J/INI.MUL
            ;2162
            ;2163 ;*
            ;2164 ; THIS ROUTINE INITIALIZES THE MULTIPLIER SEQUENCER,AND THEN TRAPS
            ;2165 ; TO THE FPA UCODE THAT PERFORMS THE MULTIPLY. IT THEN JUMPS TO THE
            ;2166 ; ROUTINE THAT READS THE PRODUCT BACK TO THE CPU.
            ;2167 ;-
            ;2168
            ;2169 =0
U 1032. 0000.003D.0180.0800.0000.10FE  ;2170 INI.MUL: CALL[TRP.CE]
```

```
ZZ-ESKAR-V22  MODULE REVISION HISTORY                                                    SEQ 2795
ESKAR65.MCR                    MICRO2  1P(00)    26-JUL-85  17:10:43                    Page   5

U 1033. 0000.003C.0180.0800.0000.1132  ;2171  NOP    ; FPA AT 0CE. INIT THE MULTIPLIER
U 1132. 0000.003C.0180.0800.0000.1034  ;2172  NOP    ; FPA AT 028.
U 1034. 0000.003D.0180.0800.0000.10FA  ;2173 =0 CALL[TRP.C8]
U 1035. 0000.003C.0180.0800.0000.1133  ;2174  NOP    ; FPA AT 0C8. START MULTIPLY
U 1133. 0000.003C.0180.0800.0000.1134  ;2175  NOP    ; FPA AT 0C9. CONTINUE MULTIPLY
U 1134. 0000.003C.0180.0800.0000.1135  ;2176  NOP    ; FPA AT 0CA. CONTINUE MULTIPLY
U 1135. 0000.003C.0180.0800.0000.1042  ;2177  J/PROD1    ; FPA AT 028.
            ;2178
            ;2179 ;*
            ;2180 ; EMODF IS THE ROUTINE THAT LOADS THE MC1,MC1 AND MP0 OPERAND
            ;2181 ; REGISTERS WITH DATA PREVIOUSLY LOADED INTO THE FAD BY THE LD.MUL
            ;2182 ; ROUTINE. IT THEN LOADS THE IB WITH AN EMODF OPCODE AND JUMPS TO
            ;2183 ; THE INI.MUL ROUTINE TO INITIALIZE AND EXECUTE THE MULTIPLY
            ;2184 ; OPERATION.
            ;2185 ;-
            ;2186
            ;2187
            ;2188
            ;2189 =0
U 1036. 2018.0039.6580.0800.4200.1119  ;2190 EMODF: ALU_K[.10],FLUSH.IB,CALL[IBLOAD]
U 1037. 0000.003C.0180.0800.0000.1038  ;2191  NOP
U 1038. 0000.003D.0180.0800.0000.1014  ;2192 =0 CALL[TRP.E3]
U 1039. 0000.003C.0180.0800.0000.103A  ;2193  NOP    ; FPA AT 0E3.MC1_AR1
U 103A. 0000.003D.0180.0800.0000.1018  ;2194 =0 CALL[TRP.E2]
U 103B. 0000.003C.0180.0800.0000.103C  ;2195  NOP    ; FPA AT 0E2. MP1_BR1
U 103C. 0000.003D.0180.0800.0000.101C  ;2196 =0 CALL[TRP.EA]
U 103D. 0000.003C.0180.0800.0000.103E  ;2197  NOP    ; FPA AT 0EA. MC1_BR0
U 103E. 2018.0039.2180.0800.4200.1119  ;2198 =0 ALU_K[.14],FLUSH.IB,CALL[IBLOAD]
U 103F. 0000.003C.0180.0800.0000.1032  ;2199  J/INI.MUL
            ;2200
            ;2201 ;*
            ;2202 ; THIS ROUTINE TRAPS TO THE FPA UCODE THAT CONTROLS THE TRANSMISSION
            ;2203 ; OF NORMALIZED AND UNNORMALIZED PRODUCTS BACK TO THE CPU FOR
            ;2204 ; VERIFICATION.
            ;2205 ;-
            ;2206
            ;2207
            ;2208
            ;2209
            ;2210
            ;2211 =0
U 1040. 0000.003D.0180.0800.0000.1010  ;2212 PROD: CALL[TRP.D1]
U 1041. 0000.003C.01D8.0800.0000.1136  ;2213  Q_ACC    ; FPA AT 0D1.
            ;2214      ; SAVE PRODUCT HI AND LO
            ;2215      ; IN AR LATCH AND IN
            ;2216      ; NORMALIZER REGISTER
            ;2217
U 1136. 0001.207C.0180.0988.0000.1137  ;2218  RC[1]_Q,CPSYNC   ; SAVE UNNORMALIZED PRODUCT HI
U 1137. 0000.003C.01D8.0800.0000.1138  ;2219  Q_ACC    ; FPA AT 0D2
U 1138. 0001.207C.0180.0980.0000.1139  ;2220  RC[0]_Q,CPSYNC   ; SAVE UNNORMALIZED PRODUCT LO
U 1139. 0000.003C.01D8.0800.0000.113A  ;2221  Q_ACC    ; FPA AT 0D3.
            ;2222      ; READ NORMALIZED PRODUCT LO
U 113A. 0A00.003C.0180.0800.0000.113B  ;2223  D_ACC    ; FPA AT 0D4,
U 113B. 0000.003E.0180.0800.0000.0001  ;2224  RETURN1    ; READ NORMALIZED PRODUCT HI
            ;2225
```

```
            ;2226 ;*
            ;2227 ; PROD1 TRAPS TO THE FPA UCODE THAT CONTROLS THE SYNCHRONIZED
            ;2228 ; TRANSMISSION OF INTEGER PRODUCT HI AND LO TO THE CPU FOR
            ;2229 ; VERIFICATION.
            ;2230 ;-
            ;2231
            ;2232
            ;2233
            ;2234
            ;2235
            ;2236 =0
U 1042. 0000,003D,0180,0800,0000,10FC  ;2237 PROD1: CALL[TRP.CC]
U 1043. 0000,007C,01D8,0800,0000,113C  ;2238  Q_ACC,CPSYNC   ; FPA AT 0CC.
            ;2239       ; BUS A_INTEGER PRODUCT HI
U 113C. 0A00,007C,0180,0800,0000,113D  ;2240  D_ACC,CPSYNC   ; FPA AT 0CD.
            ;2241       ; BUSA_INTEGER PRODUCT LO
U 113D. 0000,003C,0180,0800,0000,113E  ;2242  NOP     ; FPA AT 0CE.
            ;2243       ; INITIALIZE MULTIPLIER LOGIC
U 113E. 0000,003E,0180,0800,0000,0001  ;2244  RETURN1     ; FPA AT 028.
            ;2245
            ;2246 =0
U 1044. 0000,003D,0180,0800,0000,1116  ;2247 FAD.CHK: CALL[TRP.FB]
U 1045. 0000,003C,01D8,0800,0000,113F  ;2248  Q_ACC    ; FPA AT 0FB. BUSA_AR1
U 113F. 0001,207C,0180,09F0,0000,1140  ;2249  RC[0E]_Q,CPSYNC  ; SAVE
U 1140. 0A00,003C,0180,0800,0000,1141  ;2250  D_ACC    ; FPA AT 0FC. BUSA_AR0
U 1141. 0001,007C,0180,09E8,0000,1142  ;2251  RC[0D]_D,CPSYNC  ; SAVE
U 1142. 0000,003C,01D8,0800,0000,1143  ;2252  Q_ACC    ; FPA AT 0FD. BUSA_BR1
U 1143. 0001,207C,0180,09E0,0000,1144  ;2253  RC[0C]_Q,CPSYNC  ; SAVE
U 1144. 0A00,003C,0180,0800,0000,1145  ;2254  D_ACC    ; FPA AT 0FE. BUSA_BR0
U 1145. 0001,003C,0180,09D8,0000,1146  ;2255  RC[0B]_D   ; SAVE
U 1146. 0000,003E,0180,0800,0000,0001  ;2256  RETURN1
            ;2257
            ;2258
            ;2259 ;
            ;2260 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
            ;2261 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
            ;2262 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
            ;2263 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
            ;2264 ;
U 1147. 0000,003C,01C0,0A28,0000,1148  ;2265 UNSEQA: Q_R[5]    ; GET FRACTION INTO D AND Q.
U 1148. 0800,003C,B980,0A20,0084,7149  ;2266  D_R[4],SC_K[.19]
U 1149. 0D00,003C,0180,0800,0000,114A  ;2267  D_DAL.SC
U 114A. 0000,003C,65E0,0800,0084,714B  ;2268  Q_D,SC_K[.10]
U 114B. 0D00,003C,0180,0800,0000,114C  ;2269  D_DAL.SC
U 114C. 0001,003C,0180,0AB8,0000,114D  ;2270  R[7]_D    ; SAVE LEAST SIGNIFICANT RESULT, LO
U 114D. 0800,003C,01C0,0A20,0000,114E  ;2271  D_R[4],Q_R[4]
U 114E. 0819,0024,5980,0800,0000,114F  ;2272  ALU_D[ANDNOT]K[.7F],D_ALU
U 114F. 0100,003C,5D88,0800,0084,7150  ;2273  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 1150. 0D00,003C,0180,0800,0000,1151  ;2274  D_DAL.SC
U 1151. 0001,003C,0180,0AB0,0084,7152  ;2275  R[6]_D,SC_K[.8]
U 1152. 0818,0034,59F8,0A28,0000,1153  ;2276  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 1153. 0D00,003C,01C0,0A30,0000,1154  ;2277  D_DAL.SC,Q_R[6]
U 1154. 001D,0032,0180,0AB0,0000,0001  ;2278  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
            ;2279 ;
            ;2280 ;
```

```
            ;2281 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2282 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2283 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2284 ;
U 1156. 2000.003C.0180.0800.0000.1157  ;2285 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 1157. 3000.003C.0180.6000.0000.1158  ;2286  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 1158. 0000.003C.0180.F800.0000.1159  ;2287 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1159. 0000.003C.0180.F800.0000.115A  ;2288  MCT/ALLOW.IB.READ
U 115A. 0000.003C.0180.F800.0000.115B  ;2289  MCT/ALLOW.IB.READ
U 115B. 0000.003C.0180.F800.0000.115C  ;2290  MCT/ALLOW.IB.READ
U 115C. 0000.003C.0180.F800.0000.115D  ;2291  MCT/ALLOW.IB.READ
U 115D. 1000.003C.0180.F800.0000.115E  ;2292  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 115E. 0000.003E.0180.0800.0000.0001  ;2293  RETURN1
U 115F. 3000.003C.0180.0800.0000.1158  ;2294 WAITIB: IBC/START,J/IBW
            ;2295
            ;2296
            ;2297
            ;2298 ;
            ;2299 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;2300 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;2301 ;   28: NAD/28  ; NOP
            ;2302 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;2303 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;2304 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;2305 ;  INITIALIZE ITSELF.
            ;2306 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;2307 ;
U 1160. 0818.0038.4580.0800.0000.1161  ;2308 FPINIT: D_K[.8000]
U 1161. 0000.003C.5D80.3C00.0000.1046  ;2309  ID[ACC.CS]_D  ; TURN ON FPA
U 1046. 0818.0039.2D80.0800.0000.1165  ;2310 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1047. 0000.003C.5980.3C00.0000.1162  ;2311  ID[ACC.2]_D
U 1162. 0000.00FC.0380.0800.0000.1048  ;2312  TRAP.FPA  ; TRAP FPA TO 28.
U 1048. 0018.0039.1980.0800.0200.1156  ;2313 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 1049. 0F00.003C.0180.0800.0000.104A  ;2314  D_0
U 104A. 0000.003D.0180.0800.0000.1165  ;2315 =0 CALL,J/GENTRA
U 104B. 0000.003C.5980.3C00.0000.1163  ;2316  ID[ACC.2]_D
U 1163. 0000.00FC.0380.0800.0000.104C  ;2317  TRAP.FPA  ; TRAP FPA TO 0
U 104C. 0818.0039.2D80.0800.0000.1165  ;2318 =0 D_K[.28],CALL,J/GENTRA
U 104D. 0000.003C.5980.3C00.0000.1164  ;2319  ID[ACC.2]_D
U 1164. 0000.00FE.0380.0800.0000.0001  ;2320  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
            ;2321 ;
            ;2322 ;
            ;2323 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRESS IN THE D REGISTER AND
            ;2324 ; CONVERT IT TO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
            ;2325 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2326
U 1165. 0000.003C.65F8.0800.0084.7166  ;2327 GENTRA: SC_K[.10],Q_0
U 1166. 0D00.003C.8D80.0800.0084.7167  ;2328  D_DAL.SC,SC_K[.1F]
U 1167. 0801.001E.0180.0800.0000.0001  ;2329  ALU_D.ORNOT.MASK,D_ALU,RETURN1
            ;2330
```

```
;2331   .PAGE "TEST 299 INTEGER MULTIPLY (PRODUCT LO)"
;2332  ;*****************************************************************
;2333  ;**
;2334  ;
;2335  ; INTEGER MULTIPLY
;2336  ;
;2337  ;
;2338  ; TEST DESCRIPTION
;2339  ;
;2340  ; THIS TEST DOES A 32 BIT MULTIPLY WITH THE MULTIPLIER (MP0)
;2341  ; AND THE MULTIPLICAND INTEGER (MCI) CONTAING ZEROS. THE (MCI)
;2342  ; AND MP0 ARE THEN RE-INITIALIZED WITH (MP0) CONTAINING A
;2343  ; FLOATING ONE PATTERN AND THE MCI CONTAINING THE LSB ASSERTED.
;2344  ; UPON  COMPLETION THE TEST REVERSES THE DATA IN THE MP0 AND
;2345  ; MCI AND REPEATS THE OPERATION .THE RESULTANT PRODUCT FLOATS A
;2346  ; ONE THROUGH THE LO HALF OF THE INTEGER PRODUCT REGISTER.
;2347  ;
;2348  ;
;2349  ; TEST SEQUENCE   DESCRIPTION
;2350  ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2351  ; 2  PERFORM THE MULTIPLY
;2352  ; 3  VERIFY RESULTS
;2353  ;
;2354  ; FPA UCODE USED
;2355  ;
;2356  ; LOCATION CONTENTS
;2357  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2358  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2359  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2360  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2361  ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2362  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2363  ; 028: NAD/028
;2364  ; 0E4: FADC/B,BSC/FAL.HL,OPLD/MCI.NAD/0C8
;2365  ; 0EE: FADC/A,BSC/FAL.LH,OPLD/MP0,NAD/0C8
;2366  ; 028: NAD/028
;2367  ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2368  ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2369  ; 028: NAD/028
;2370  ; 0CC: BSC/INTH,FPSYNC,WAIT.NAD/0CD
;2371  ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;2372  ; 0CE: OPLD/INIT,NAD/028
;2373  ;
;2374  ; LOGIC DESCRIPTION
;2375  ;
;2376  ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2377  ; LOGIC ARE ON THE FML AND FMH MODULES.
;2378  ;
;2379  ;
;2380  ;
;2381  ;
;2382  ; ERROR DESCRIPTION
;2383  ;
;2384  ; EXPECTED INTEGER PRODUCT HI
;2385  ; RECEIVED INTEGER PRODUCT HI
```

```
            ;2386 ; EXPECTED INTEGER PRODUCT LO
            ;2387 ; RECEIVED INTEGER PRODUCT LO
            ;2388 ; DATA IN MP0
            ;2389 ; DATA IN MCI
            ;2390 ; PASS FLAG
            ;2391 ;
            ;2392 ;
            ;2393 ;
            ;2394 ; SYNC POINT
            ;2395 ;
            ;2396 ;
            ;2397 ;
            ;2398 ;--
            ;2399 ;
            ;2400 ;*****************************************************************
            ;2401
            ;2402
            ;2403 =0
U 1050, 0018,0039,5D80,09F8,0000,10D3  ;2404 MULT3: NEWTST[.7]
            ;2405
            ;2406 ; PERFORM 0X0 MULTIPLICATION
            ;2407
U 1051, 0000,003C,0180,0800,0000,1052  ;2408  NOP
U 1052, 0000,003D,0180,0800,0000,10E5  ;2409 =0 ERLOOP
U 1053, 0003,003C,0180,0AB8,0000,1168  ;2410  R[7]_0    ; MP0_0
U 1168, 0003,003C,0180,09D0,0000,1169  ;2411  RC[0A]_0   ; SAVE
U 1169, 0003,003C,0180,0AC8,0000,116A  ;2412  R[9]_0    ; MCI_0
U 116A, 0003,003C,0180,09E0,0000,116B  ;2413  RC[0C]_0   ; SAVE
U 116B, 0003,003C,0180,09F0,0000,1054  ;2414  RC[0E]_0
U 1054, 0000,003D,0180,0800,0000,1020  ;2415 =0 CALL[LD.MUL]
U 1055, 0000,003C,0180,0800,0000,1056  ;2416  NOP
U 1056, 0000,003D,0180,0800,0000,1028  ;2417 =0 CALL[MULL]
U 1057, 0001,203C,0180,09E8,0000,116C  ;2418  RC[0D]_Q   ; SAVE RECEIVED INTEGER PRODUCT HI
U 116C, 0001,003C,0180,09D8,0000,116D  ;2419  RC[0B]_D   ; SAVE RECEIVED INTEGER PRODUCT LO
            ;2420
            ;2421 ; VERIFY RESULTS
            ;2422
U 116D, 0011,2020,0180,0970,0010,116E  ;2423  ALU_Q.XOR.RC[0E],CLK.UBCC ; VERIFY INTEGER PRODUCT HI
U 116E, 0000,013C,0180,0800,0000,1058  ;2424  Z?
U 1058, 0000,003D,0180,0800,0000,10E6  ;2425 =0 ERROR1
U 1059, 0010,0038,01C0,0958,0000,116F  ;2426  Q_RC[0B]
U 116F, 0011,2020,0180,0960,0010,1170  ;2427  ALU_Q.XOR.RC[0C],CLK.UBCC ; VERIFY INTEGER PRODUCT LO
U 1170, 0000,013C,0180,0800,0000,105C  ;2428  Z?
U 105C, 0000,003D,0180,0800,0000,10E6  ;2429 =0 ERROR1
U 105D, 0001,003C,0180,09D8,0000,1171  ;2430  RC[0B]_D
            ;2431
            ;2432 ; RE-INITIALIZE FOR FLOATING ONE BY (1) MULTIPLICATION.
            ;2433
U 1171, 0018,0038,0580,09C0,0000,1172  ;2434  RC[8]_K[.1]    ; PASS FLAG.
            ;2435       ; 1= MP0_PATTERN,MCI_LSB SET
            ;2436       ; 0= MCI_PATTERN,MP0_LSB SET
U 1172, 0018,0038,0580,0AB8,0000,1173  ;2437  R[7]_K[.1]    ; MP0 DATA
U 1173, 0800,003C,0180,0A38,0000,1174  ;2438  D_R[7]
U 1174, 0001,003C,0180,09D0,0000,1175  ;2439  RC[0A]_D    ; SAVE
U 1175, 0001,003C,0180,09E0,0000,1176  ;2440  RC[0C]_D    ; EXPECTED INT.PROD.LO
```

```
U 1176. 0001.003C.0180.0AB0.0000.1177  ;2441  R[6]_D     ; MP1 DATA (MP1 NOT SELECTED)
U 1177. 0001.003C.0180.0AC0.0000.1178  ;2442  R[8]_D     ; MC1 DATA (MC1 NOT SELECTED)
U 1178. 0001.003C.0180.09C8.0000.1179  ;2443  RC[9]_D     ; SAVE
U 1179. 0003.003C.0180.09F0.0000.117A  ;2444  RC[0E]_0    ; EXPECTED INT. PROD. HI
U 117A. 0810.0038.0180.0950.0000.117B  ;2445 MUL3: D_RC[0A]    ;
U 117B. 0001.003C.0180.0AB8.0000.117C  ;2446  R[7]_D     ; DATA FOR MP0
U 117C. 0810.0038.0180.0948.0000.117D  ;2447  D_RC[9]
U 117D. 0001.003C.0180.0AC8.0000.1060  ;2448  R[9]_D     ; DATA FOR MC1
          ;2449
          ;2450 ;  MULTIPLY, SAVE PRODUCT
          ;2451
U 1060. 0000.003D.0180.0800.0000.1020  ;2452 =0 CALL[LD.MUL]    ; LOAD FAD A,B LATCHES
U 1061. 0000.003C.0180.0800.0000.1062  ;2453  NOP
U 1062. 0000.003D.0180.0800.0000.1028  ;2454 =0 CALL[MULL]    ; EXECUTE MULL R,R
U 1063. 0001.203C.0180.09E8.0000.117E  ;2455  RC[0D]_Q    ; SAVE RECEIVED INT. PROD. HI
U 117E. 0001.003C.0180.09D8.0000.117F  ;2456  RC[0B]_D    ; SAVE RECEIVED INT. PROD. LO
          ;2457
          ;2458 ; VERIFY RESULTS
          ;2459
U 117F. 0011.2020.0180.0970.0010.1180  ;2460  ALU_Q.XOR.RC[0E],CLK.UBCC ; VERIFY INTEGER PRODUCT HI
U 1180. 0000.013C.0180.0800.0000.1064  ;2461  Z?
U 1064. 0000.003D.0180.0800.0000.10E6  ;2462 =0 ERROR1
U 1065. 0010.0038.01C0.0958.0000.1181  ;2463  Q_RC[0B]
U 1181. 0011.2020.0180.0960.0010.1182  ;2464  ALU_Q.XOR.RC[0C],CLK.UBCC ; VERIFY INTEGER PRODUCT LO
U 1182. 0000.013C.0180.0800.0000.1066  ;2465  Z?
U 1066. 0000.003D.0180.0800.0000.10E6  ;2466 =0 ERROR1
U 1067. 0810.0038.0180.0940.0010.1183  ;2467  D_RC[8],CLK.UBCC
U 1183. 0000.013C.0180.0800.0000.1068  ;2468  Z?
U 1068. 0000.003C.0180.0800.0000.1184  ;2469 =0 J/FLG1     ; MP0 GETS PATTERN
U 1069. 0000.003C.0180.0800.0000.118A  ;2470  J/FLG2     ; MC1 GETS PATTERN
U 1184. 0810.0038.0180.0950.0000.1185  ;2471 FLG1: D_RC[0A]
U 1185. 0021.003C.0180.09E0.0000.1186  ;2472  ALU_D,R̄C[0C]_ALU.LEFT  ; UPDATE EXPECTED INT. PROD. LO
U 1186. 0000.0D3C.0180.0800.0000.1006  ;2473  D31?
U 1006. 0021.003C.0180.09D0.0000.117A  ;2474 =110 ALU_D,RC[0A]_ALU.LEFT,J/MUL3 ; UPDATE PATTERN
U 1007. 0018.0038.0580.09E0.0000.1187  ;2475  RC[0C]_K[.1]    ; EXPECTED INTEGER PRODUCT LO
U 1187. 0003.003C.0180.09C0.0000.1188  ;2476  RC[8]_0     ; SET FLAG FOR MC1 GETS PATTERN
U 1188. 0018.0038.0580.09C8.0000.1189  ;2477  RC[9]_K[.1]    ; MC1 GETS FLOATING (1)
U 1189. 0018.0038.0580.09D0.0000.117A  ;2478  RC[0A]_K[.1],J/MUL3  ; MP0 GETS LSB=1
U 118A. 0810.0038.0180.0948.0000.118B  ;2479 FLG2: D_RC[9]
U 118B. 0021.003C.0180.09E0.0000.118C  ;2480  ALU_D,R̄C[0C]_ALU.LEFT  ; UPDATE EXPECTED INT. PROD. LO
U 118C. 0000.0D3C.0180.0800.0000.1016  ;2481  D31?
U 1016. 0021.003C.0180.09C8.0000.117A  ;2482 =110 ALU_D,RC[9]_ALU.LEFT,J/MUL3 ; UPDATE PATTERN
U 1017. 0000.003C.0180.0800.0000.106A  ;2483  NOP
          ;2484
          ;2485
          ;2486
          ;2487
```

```
;2488  .PAGE "TEST 29A INTEGER MULTIPLY (PRODUCT HI)"
;2489 ;*****************************************************************
;2490 ;**
;2491 ; INTEGER MULTIPLY
;2492 ;
;2493 ; TEST DESCRIPTION
;2494 ;
;2495 ; THIS TEST GENERATES MPO AND MCI OPERANDS THAT YIELD A FLOATING
;2496 ; (1) PRODUCT IN THE INTEGER PRODUCT HI. THE MPO MSB IS ASSERTED.
;2497 ; THE MCI RECEIVES A FLOATING ONE PATTERN THAT STARTS AT BIT
;2498 ; POSITION (1).
;2499 ;
;2500 ; TEST SEQUENCE   DESCRIPTION
;2501 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2502 ; 2  PERFORM THE MULTIPLY
;2503 ; 3  VERIFY RESULTS
;2504 ;
;2505 ; FPA UCODE USED
;2506 ;
;2507 ; LOCATION CONTENTS
;2508 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2509 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2510 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2511 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2512 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2513 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2514 ; 028: NAD/028
;2515 ; 0E4: FADC/B,BSC/FAL.HL,OPLD/MCI.NAD/0C8
;2516 ; 0EE: FADC/A,BSC/FAL.LH,OPLD/MPO,NAD/0C8
;2517 ; 028: NAD/028
;2518 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2519 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2520 ; 028: NAD/028
;2521 ; 0CC: BSC/INTH,FPSYNC,WAIT.NAD/0CD
;2522 ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;2523 ; 0CE: OPLD/INIT,NAD/028
;2524 ;
;2525 ; LOGIC DESCRIPTION
;2526 ;
;2527 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2528 ; LOGIC ARE ON THE FML AND FMH MODULES.
;2529 ;
;2530 ;
;2531 ;
;2532 ;
;2533 ; ERROR DESCRIPTION
;2534 ;
;2535 ; EXPECTED INTEGER PRODUCT HI
;2536 ; RECEIVED INTEGER PRODUCT HI
;2537 ; EXPECTED INTEGER PRODUCT LO
;2538 ; RECEIVED INTEGER PRODUCT LO
;2539 ; DATA IN MPO
;2540 ; DATA IN MCI
;2541 ;
;2542 ;
```

```
            ;2543 ;
            ;2544 ; SYNC POINT
            ;2545 ;
            ;2546 ;
            ;2547 ;
            ;2548 ;
            ;2549 ;-
            ;2550 ;
            ;2551 ;*****************************************************************
            ;2552
            ;2553
            ;2554 ;/////////////////////////////////////////////////
            ;2555 ;*
            ;2556 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER MSB SET.
            ;2557
            ;2558
            ;2559 =0
U 106A. 0018.0039.5D80.09F8.0000.10D3  ;2560 MULT4: NEWTST[.7]
U 106B. 0818.0038.0980.0800.0000.118D  ;2561  D_K[.2]    ; MP0 DATA
U 118D. 0001.003C.0180.0AC8.0000.118E  ;2562  R[9]_D
U 118E. 0001.003C.0180.09C8.0000.118F  ;2563  RC[9]_D    ; SAVE
U 118F. 0600.003C.0180.0800.0000.1190  ;2564  D_D.RIGHT
U 1190. 0001.003C.0180.09F0.0000.1191  ;2565  RC[0E]_D   ; EXPECTED INT.PROD. HI
U 1191. 0003.003C.0180.09E0.0000.1192  ;2566  RC[0C]_0   ; EXPECTED INT.PROD.LO
U 1192. 0001.003C.0180.0AB0.0000.1193  ;2567  R[6]_D    ; MP1 DATA (MP1 NOT SELECTED)
U 1193. 0001.003C.0180.0AC0.0000.1194  ;2568  R[8]_D    ; MC1 DATA (MC1 NOT SELECTED)
U 1194. 0000.003C.8D80.0800.0084.7195  ;2569  SC_K[.1F]
U 1195. 0803.001C.0180.0800.0000.1196  ;2570  D_NOT.MASK
U 1196. 0001.003C.0180.0AB8.0000.1197  ;2571  R[7]_D    ; SET MSB IN MP0
U 1197. 0001.003C.0180.09D0.0000.106C  ;2572  RC[0A]_D   ; SAVE
U 106C. 0000.003D.0180.0800.0000.10E5  ;2573 =0 ERLOOP
U 106D. 0810.0038.0180.0948.0000.1198  ;2574 MUL4: D_RC[9]
U 1198. 0001.003C.0180.0AC8.0000.106E  ;2575  R[9]_D    ; DATA FOR MC1
            ;2576
            ;2577
            ;2578 ;  MULTIPLY, SAVE PRODUCT
            ;2579
U 106E. 0000.003D.0180.0800.0000.1020  ;2580 =0 CALL[LD.MUL]  ; LOAD FAD A,B LATCHES
U 106F. 0000.003C.0180.0800.0000.1070  ;2581  NOP
U 1070. 0000.003D.0180.0800.0000.1028  ;2582 =0 CALL[MULL]  ; EXECUTE MULL R,R
U 1071. 0001.203C.0180.09E8.0000.1199  ;2583  RC[0D]_Q   ; SAVE RECEIVED INT. PROD. HI
U 1199. 0001.003C.0180.09D8.0000.119A  ;2584  RC[0B]_D   ; SAVE RECEIVED INT. PROD. LO
            ;2585
            ;2586 ; VERIFY RESULTS
            ;2587
U 119A. 0011.2020.0180.0970.0010.119B  ;2588  ALU_Q.XOR.RC[0E],CLK.UBCC
U 119B. 0000.013C.0180.0800.0000.1072  ;2589  Z?
U 1072. 0000.003D.0180.0800.0000.10E6  ;2590 =0 ERROR1
U 1073. 0010.0038.01C0.0958.0000.119C  ;2591  Q_RC[0B]
U 119C. 0011.2020.0180.0960.0010.119D  ;2592  ALU_Q.XOR.RC[0C],CLK.UBCC
U 119D. 0000.013C.0180.0800.0000.1074  ;2593  Z?
U 1074. 0000.003D.0180.0800.0000.10E6  ;2594 =0 ERROR1
U 1075. 0810.0038.0180.0948.0000.119E  ;2595  D_RC[9]
U 119E. 0001.003C.0180.09F0.0000.119F  ;2596  RC[0E]_D   ; UPDATE EXPECTED PROD. HI
U 119F. 0000.0D3C.0180.0800.0000.1026  ;2597  D31?
```

```
U 1026, 0021,003C,0180,09C8,0000,106D  ;2598 =110 ALU_D,RC[9]_ALU.LEFT,J/MUL4
U 1027, 0000,003C,0180,0800,0000,1076  ;2599  NOP
            ;2600
```

```
;2601   .PAGE "TEST 29B MULTIPLY FLOAT, FLOATING ONE BY 0.1"
;2602  ;*************************************************************
;2603  ;**
;2604  ;
;2605  ; MULTIPLY FLOAT, FLOATING ONE BY  0.1
;2606  ;
;2607  ;
;2608  ; TEST DESCRIPTION
;2609  ;
;2610  ; THIS TEST DOES A MUL FLOAT WITH THE MULTIPLICAND (MC1)
;2611  ; CONTAINING A FLOATING ONE PATTERN AND THE MULTIPLIER
;2612  ; (MP1) CONTAINING ONLY THE HIDDEN BIT SET.
;2613  ;
;2614  ;
;2615  ; TEST SEQUENCE   DESCRIPTION
;2616  ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2617  ; 2  PACK THE DATA
;2618  ; 3  PERFORM THE MULTIPLY
;2619  ; 4  VERIFY RESULTS
;2620  ;
;2621  ; FPA UCODE USED
;2622  ;
;2623  ; LOCATION CONTENTS
;2624  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2625  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2626  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2627  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2628  ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2629  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2630  ; 028: NAD/028
;2631  ; 0E1: FADC/A,BSC/FALHL,OPLD/MC1,NAD/0E2
;2632  ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;2633  ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2634  ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2635  ; 028: NAD/028
;2636  ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2637  ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2638  ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2639  ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2640  ;
;2641  ; LOGIC DESCRIPTION
;2642  ;
;2643  ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2644  ; LOGIC ARE ON THE FML AND FMH MODULES.
;2645  ;
;2646  ;
;2647  ;
;2648  ;
;2649  ;
;2650  ; ERROR DESCRIPTION
;2651  ;
;2652  ; PACKED EXPECTED NORMALIZED PRODUCT HI
;2653  ; PACKED RECEIVED NORMALIZED PRODUCT HI
;2654  ; PACKED EXPECTED NORMALIZED PRODUCT LO
;2655  ; PACKED RECEIVED NORMALIZED PRODUCT LO
```

```
            ;2656 ; PACKED EXPECTED UNNORMALIZED PRODUCT HI
            ;2657 ; PACKED RECEIVED UNNORMALIZED PRODUCT HI
            ;2658 ; PACKED EXPECTED UNNORMALIZED PRODUCT LO
            ;2659 ; PACKED RECEIVED UNNORMALIZED PRODUCT LO
            ;2660 ; UNPACKED DATA HI
            ;2661 ; UNPACKED DATA LO
            ;2662 ; MULTIPLIER REGISTER HI (MP1)
            ;2663 ; MULTIPLICAND REGISTER HI (MC1)
            ;2664 ; BIT COUNTER
            ;2665 ;
            ;2666 ;
            ;2667 ;
            ;2668 ; SYNC POINT
            ;2669 ;
            ;2670 ;
            ;2671 ;
            ;2672 ;--
            ;2673 ;
            ;2674 ;*******************************************************
            ;2675
            ;2676
            ;2677 ;/////////////////////////////////////////////////////
            ;2678 ;*
            ;2679 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
            ;2680
            ;2681
            ;2682 =0
U 1076, 0018.0039.8980.09F8.0000.10D3  ;2683 MULT1: NEWTST[.D]
U 1077, 0818.0038.4D80.0800.0000.11A0  ;2684  D_K[.FF00]
U 11A0, 0819.0030.4180.0800.0000.11A1  ;2685  D_D.OR.K[.80]
U 11A1, 0001.0028.0180.0A88.0000.11A2  ;2686  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 11A2, 0000.003C.6180.0800.0084.71A3  ;2687  SC_K[.F]
U 11A3, 0000.003C.01F8.0800.0000.11A4  ;2688  Q_0
U 11A4, 0001.2030.0180.0A80.0000.11A5  ;2689  ALU_Q.OR.MASK,R[0]_ALU ; MASK OUT BIT 15
U 11A5, 0000.003C.0580.0800.0084.B1A6  ;2690  SC_SC-K[.1]
U 11A6, 0801.0034.0180.0800.0000.11A7  ;2691  ALU_D.AND.MASK,D_ALU
U 11A7, 0001.0028.0180.0A90.0000.11A8  ;2692  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 11A8, 0818.0038.7980.0800.0000.11A9  ;2693  D_K[.30]
U 11A9, 0819.0030.8980.0800.0000.11AA  ;2694  D_D.OR.K[.D]
U 11AA, 0001.003C.0180.0990.0000.1078  ;2695  RC[2]_D   ; BIT COUNT OF 61
U 1078, 0000.003D.0180.0800.0000.10E5  ;2696 =0 ERLOOP
U 1079, 0810.0038.0180.0910.0082.11AB  ;2697 MUL1: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 11AB, 0003.003C.0180.0AA8.0000.11AC  ;2698  R[5]_0   ; MULTIPLIER LO GETS 0
U 11AC, 0003.001C.0180.0AA0.0000.11AD  ;2699  R[4]_NOT.MASK  ; FLOATING 1
            ;2700
            ;2701 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;2702
U 11AD, 0800.003C.0180.0A20.0000.11AE  ;2703  D_R[4]
U 11AE, 0001.003C.0180.09A0.0000.11AF  ;2704  RC[4]_D   ; SAVE HI
U 11AF, 0800.003C.0180.0A28.0000.11B0  ;2705  D_R[5]
U 11B0, 0001.003C.0180.0998.0000.11B1  ;2706  RC[3]_D   ; SAVE LO
U 11B1, 0000.003C.01C0.0A20.0000.11B2  ;2707  Q_R[4]
U 11B2, 0800.003C.0180.0A28.0000.11B3  ;2708  D_R[5]
U 11B3, 0000.003C.00B0.0800.0000.11B4  ;2709  Q_Q.RIGHT.SI/1
U 11B4, 0600.003C.0180.0800.0000.11B5  ;2710  D_D.RIGHT.SI/ZERO
```

```
U 11B5. 0001.203C.0180.0AA0.0000.11B6  ;2711  R[4]_Q
U 11B6. 0001.003C.0180.0AA8.0000.107A  ;2712  R[5]_D
          ;2713
          ;2714 ; PACK THE DATA
          ;2715
U 107A. 0000.003D.0180.0800.0000.1147  ;2716 =0 CALL[UNSEQA]  ; PACK UNNORMALIZED EXPECTED PRODUCT
U 107B. 0800.003C.0180.0A30.0000.11B7  ;2717  D_R[6]
U 11B7. 080D.0034.0180.0A10.0000.11B8  ;2718  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11B8. 0001.003C.0180.0AB0.0000.11B9  ;2719  R[6]_D
U 11B9. 0819.0030.3180.0800.0000.11BA  ;2720  D_D.OR.K[.40]
U 11BA. 0001.003C.0180.09D0.0000.11BB  ;2721  RC[0A]_D  ; SAVE HI
U 11BB. 0800.003C.0180.0A38.0000.11BC  ;2722  D_R[7]
U 11BC. 0001.003C.0180.09C0.0000.11BD  ;2723  RC[8]_D   ; SAVE LO
          ;2724
          ;2725 ; GENERATE NORMALIZED EXPECTED PRODUCT
          ;2726
U 11BD. 0810.0038.0180.0920.0000.11BE  ;2727  D_RC[4]   ;  HI
U 11BE. 0001.003C.0180.0AA0.0000.11BF  ;2728  R[4]_D
U 11BF. 0810.0038.0180.0918.0000.11C0  ;2729  D_RC[3]   ;  LO
U 11C0. 0018.0038.31C0.0800.0000.11C1  ;2730  Q_K[.40]
U 11C1. 001D.2014.0180.0AA8.0010.11C2  ;2731  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 11C2. 0000.1B3C.0180.0800.0000.101E  ;2732  ALU.C?
U 101E. 0000.003C.0180.0800.0000.107C  ;2733 =1110 J/PAK2
U 101F. 0800.003C.0180.0A20.0000.11C3  ;2734  D_R[4]
U 11C3. 0019.0014.0580.0AA0.0000.107C  ;2735  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
          ;2736
          ;2737 ; PACK NORMALIZED EXPECTED PRODUCT
          ;2738
          ;2739 =0
U 107C. 0000.003D.0180.0800.0000.1147  ;2740 PAK2: CALL[UNSEQA]  ; PACK NORMALIZED EXPECTED PRODUCT
U 107D. 0800.003C.0180.0A30.0000.11C4  ;2741  D_R[6]
U 11C4. 080D.0034.0180.0A08.0000.11C5  ;2742  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11C5. 0001.003C.0180.0AB0.0000.11C6  ;2743  R[6]_D
U 11C6. 0001.003C.0180.09F0.0000.11C7  ;2744  RC[0E]_D  ; SAVE NORMALIZED EXPECTED PRODUCT HI
U 11C7. 0800.003C.0180.0A38.0000.11C8  ;2745  D_R[7]
U 11C8. 0001.003C.0180.09E0.0000.11C9  ;2746  RC[0C]_D  ; SAVE NORMALIZED EXPECTED PRODUCT LO
          ;2747
          ;2748 ; PACK OPERANDS FOR MULTIPLY OPERATION
          ;2749
U 11C9. 0810.0038.0180.0920.0000.11CA  ;2750  D_RC[4]
U 11CA. 0001.003C.0180.0AA0.0000.11CB  ;2751  R[4]_D   ; RESTORE  HI
U 11CB. 0810.0038.0180.0918.0000.11CC  ;2752  D_RC[3]
U 11CC. 0001.003C.0180.0AA8.0000.107E  ;2753  R[5]_D   ; RESTORE  LO
U 107E. 0000.003D.0180.0800.0000.1147  ;2754 =0 CALL[UNSEQA]
U 107F. 0800.003C.0180.0A30.0000.11CD  ;2755  D_R[6]
U 11CD. 080D.0034.0180.0A08.0000.11CE  ;2756  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11CE. 0003.003C.0180.0AC0.0000.11CF  ;2757  R[8]_0   ; MULTIPLIER HI MASKED
U 11CF. 0003.003C.0180.09B0.0000.11D0  ;2758  RC[6]_0   ; SAVE
U 11D0. 0001.003C.0180.0AB0.0000.11D1  ;2759  R[6]_D   ; MULTIPLICAND HI
U 11D1. 0001.003C.0180.0AB8.0000.11D2  ;2760  R[7]_D
U 11D2. 0001.003C.0180.0AC8.0000.11D3  ;2761  R[9]_D   ; MULTIPLIER LO
U 11D3. 0001.003C.0180.09A8.0000.1080  ;2762  RC[5]_D   ; SAVE
          ;2763
          ;2764 ; LOAD OPERANDS, MULTIPLY.
          ;2765
```

```
U 1080, 0000,003D,0180,0800,0000,1020  ;2766 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
U 1081, 0000,003C,0180,0800,0000,1082  ;2767  NOP
U 1082, 0000,003D,0180,0800,0000,1022  ;2768 =0 CALL[MULF]  ; EXECUTE MULF R,R
U 1083, 081C,2034,0180,0A08,0000,11D4  ;2769  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 11D4, 0001,003C,0180,09E8,0000,11D5  ;2770  RC[0D]_D  ; SAVE NORMALIZED PRODUCT HI
U 11D5, 0810,0038,0180,0908,0000,11D6  ;2771  D_RC[1]
U 11D6, 080D,0034,0180,0A10,0000,11D7  ;2772  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11D7, 0001,003C,0180,09C8,0000,11D8  ;2773  RC[9]_D  ; SAVE UNNORMALIZED PRODUCT HI
U 11D8, 001C,0034,01C0,0A00,0000,11D9  ;2774  Q_Q.AND.R[0]  ; MASK OUT ROUND BIT
U 11D9, 0001,203C,0180,09D8,0000,11DA  ;2775  RC[0B]_Q  ; SAVE MASKED NORMALIZED PRODUCT LO
U 11DA, 0000,003C,0180,0900,0000,11DB  ;2776  LC_RC[0]
U 11DB, 0010,0038,0180,09B8,0000,11DC  ;2777  RC[7]_LC  ; SAVE UNNORMALIZED PRODUCT LO
U 11DC, 0011,2020,0180,0960,0010,11DD  ;2778  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED PRODUCT LO
U 11DD, 0000,013C,0180,0800,0000,1084  ;2779  Z?
U 1084, 0000,003D,0180,0800,0000,10E6  ;2780 =0 ERROR1
U 1085, 0010,0038,01C0,0968,0000,11DE  ;2781  Q_RC[0D]
U 11DE, 0011,2020,0180,0970,0010,11DF  ;2782  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED PRODUCT HI
U 11DF, 0000,013C,0180,0800,0000,1086  ;2783  Z?
U 1086, 0000,003D,0180,0800,0000,10E6  ;2784 =0 ERROR1
U 1087, 0010,0038,01C0,0940,0000,11E0  ;2785  Q_RC[8]
U 11E0, 0011,2020,0180,0938,0010,11E1  ;2786  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 11E1, 0000,013C,0180,0800,0000,1088  ;2787  Z?
U 1088, 0000,003D,0180,0800,0000,10E6  ;2788 =0 ERROR1
U 1089, 0010,0038,01C0,0950,0000,11E2  ;2789  Q_RC[0A]
U 11E2, 0011,2020,0180,0948,0010,11E3  ;2790  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 11E3, 0000,013C,0180,0800,0000,108A  ;2791  Z?
U 108A, 0000,003D,0180,0800,0000,10E6  ;2792 =0 ERROR1
U 108B, 0810,0038,0180,0910,0000,11E4  ;2793  D_RC[2]
U 11E4, 0018,0038,2DC0,0800,0000,11E5  ;2794  Q_K[.28]
U 11E5, 001D,0000,0180,0800,0010,11E6  ;2795  ALU_D-Q,CLK.UBCC  ; CHECK BIT COUNTER
U 11E6, 0000,013C,0180,0800,0000,108C  ;2796  Z?  ; 24 BITS PROCESSED?
U 108C, 0019,0000,0580,0990,0000,1079  ;2797 =0 ALU_D-K[.1],RC[2]_ALU,J/MUL1 ; DECREMENT BIT COUNTER
U 108D, 0000,003C,0180,0800,0000,108E  ;2798  NOP
          ;2799
```

```
;2800  .PAGE "TEST 29C MULTIPLY FLOAT, 0.1 BY  FLOATING ONE "
;2801 ;*****************************************************************
;2802 ;**
;2803 ;
;2804 ; MULTIPLY FLOAT, 0.1 BY  FLOATING ONE
;2805 ;
;2806 ;
;2807 ; TEST DESCRIPTION
;2808 ;
;2809 ; THIS TEST DOES A MUL FLOAT WITH THE MULTIPLIER (MP1)
;2810 ; CONTAINING A FLOATING ONE PATTERN AND THE MULTIPLICAND
;2811 ; (MC1) CONTAINING ONLY THE HIDDEN BIT SET.
;2812 ;
;2813 ;
;2814 ; TEST SEQUENCE   DESCRIPTION
;2815 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2816 ; 2  PACK THE DATA
;2817 ; 3  PERFORM THE MULTIPLY
;2818 ; 4  VERIFY RESULTS
;2819 ;
;2820 ; FPA UCODE USED
;2821 ;
;2822 ; LOCATION CONTENTS
;2823 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2824 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2825 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2826 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2827 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2828 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2829 ; 028: NAD/028
;2830 ; 0E1: FADC/A,BSC/FALHL,OPLD/MC1,NAD/0E2
;2831 ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;2832 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2833 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2834 ; 028: NAD/028
;2835 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2836 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2837 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2838 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2839 ;
;2840 ; LOGIC DESCRIPTION
;2841 ;
;2842 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2843 ; LOGIC ARE ON THE FML AND FMH MODULES.
;2844 ;
;2845 ;
;2846 ;
;2847 ;
;2848 ;
;2849 ; ERROR DESCRIPTION
;2850 ;
;2851 ; PACKED EXPECTED NORMALIZED PRODUCT HI
;2852 ; PACKED RECEIVED NORMALIZED PRODUCT HI
;2853 ; PACKED EXPECTED NORMALIZED PRODUCT LO
;2854 ; PACKED RECEIVED NORMALIZED PRODUCT LO
```

```
            ;2855 ; PACKED EXPECTED UNNORMALIZED PRODUCT HI
            ;2856 ; PACKED RECEIVED UNNORMALIZED PRODUCT HI
            ;2857 ; EXPECTED UNNORMALIZED PRODUCT LO
            ;2858 ; RECEIVED UNNORMALIZED PRODUCT LO
            ;2859 ; MULTIPLIER REGISTER HI
            ;2860 ; MULTIPLIER REGISTER LO
            ;2861 ; MULTIPLICAND REGISTER HI
            ;2862 ; MULTIPLICAND REGISTER LO
            ;2863 ; BIT COUNTER
            ;2864 ;
            ;2865 ;
            ;2866 ;
            ;2867 ; SYNC POINT
            ;2868 ;
            ;2869 ;
            ;2870 ;
            ;2871 ;--
            ;2872 ;
            ;2873 ;*************************************************************************
            ;2874
            ;2875
            ;2876
            ;2877 ;/////////////////////////////////////////////////////
            ;2878 ;*
            ;2879 ; FLOAT A 1 THROUGH THE MULTIPLIER WITH MULTIPLICAND=0.1.
            ;2880
            ;2881
            ;2882 =0
U 108E, 0018.0039.8980.09F8.0000.10D3  ;2883 MULT2: NEWTST[.D]
U 108F, 0818.0038.4D80.0800.0000.11E7  ;2884  D_K[.FF00]
U 11E7, 0819.0030.4180.0800.0000.11E8  ;2885  D_D.OR.K[.80]
U 11E8, 0001.0028.0180.0A88.0000.11E9  ;2886  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 11E9, 0000.003C.6180.0800.0084.71EA  ;2887  SC_K[.F]
U 11EA, 0000.003C.01F8.0800.0000.11EB  ;2888  Q_0
U 11EB, 0001.2030.0180.0A80.0000.11EC  ;2889  ALU_Q.OR.MASK,R[0]_ALU ; MASK OUT BIT 15
U 11EC, 0000.003C.0580.0800.0084.B1ED  ;2890  SC_SC-K[.1]
U 11ED, 0801.0034.0180.0800.0000.11EE  ;2891  ALU_D.AND.MASK,D_ALU
U 11EE, 0001.0028.0180.0A90.0000.11EF  ;2892  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 11EF, 0818.0038.7980.0800.0000.11F0  ;2893  D_K[.30]
U 11F0, 0819.0030.8980.0800.0000.11F1  ;2894  D_D.OR.K[.D]
U 11F1, 0001.003C.0180.0990.0000.1090  ;2895  RC[2]_D  ; BIT COUNT OF 61
U 1090, 0000.003D.0180.0800.0000.10E5  ;2896 =0 ERLOOP
U 1091, 0810.0038.0180.0910.0082.11F2  ;2897 MUL2: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 11F2, 0003.003C.0180.0AA8.0000.11F3  ;2898  R[5]_0  ; MULTIPLIER LO GETS 0
U 11F3, 0003.001C.0180.0AA0.0000.11F4  ;2899  R[4]_NOT.MASK  ; FLOATING 1
            ;2900
            ;2901 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;2902
U 11F4, 0800.003C.0180.0A20.0000.11F5  ;2903  D_R[4]
U 11F5, 0001.003C.0180.09B0.0000.11F6  ;2904  RC[6]_D  ; SAVE HI
U 11F6, 0800.003C.0180.0A28.0000.11F7  ;2905  D_R[5]
U 11F7, 0001.003C.0180.09A8.0000.11F8  ;2906  RC[5]_D  ; SAVE LO
U 11F8, 0000.003C.01C0.0A20.0000.11F9  ;2907  Q_R[4]
U 11F9, 0800.003C.0180.0A28.0000.11FA  ;2908  D_R[5]
U 11FA, 0600.003C.0080.0800.0000.11FB  ;2909  D_D.RIGHT,SI/1
```

```
U 11FB. 0000.003C.01B0.0800.0000.11FC  ;2910  Q_Q.RIGHT,SI/ZERO
U 11FC. 0001.203C.0180.0AA0.0000.11FD  ;2911  R[4]_Q
U 11FD. 0001.003C.0180.0AA8.0000.1092  ;2912  R[5]_D
            ;2913
            ;2914 ; PACK THE DATA
            ;2915
U 1092. 0000.003D.0180.0800.0000.1147  ;2916 =0 CALL[UNSEQA]  ; PACK UNNORMALIZED EXPECTED PRODUCT
U 1093. 0800.003C.0180.0A30.0000.11FE  ;2917  D_R[6]
U 11FE. 080D.0034.0180.0A10.0000.11FF  ;2918  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11FF. 0001.003C.0180.0AB0.0000.1200  ;2919  R[6]_D
U 1200. 0819.0030.3180.0800.0000.1201  ;2920  D_D.OR.K[.40]
U 1201. 0001.003C.0180.09D0.0000.1202  ;2921  RC[0A]_D  ; SAVE
U 1202. 0800.003C.0180.0A38.0000.1203  ;2922  D_R[7]
U 1203. 0001.003C.0180.09C0.0000.1204  ;2923  RC[8]_D   ; SAVE
            ;2924
            ;2925 ; GENERATE NORMALIZED EXPECTED PRODUCT
            ;2926
U 1204. 0810.0038.0180.0930.0000.1205  ;2927  D_RC[6]   ;  HI
U 1205. 0001.003C.0180.0AA0.0000.1206  ;2928  R[4]_D
U 1206. 0810.0038.0180.0928.0000.1207  ;2929  D_RC[5]   ;  LO
U 1207. 0018.0038.31C0.0800.0000.1208  ;2930  Q_K[.40]
U 1208. 001D.2014.0180.0AA8.0010.1209  ;2931  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 1209. 0000.1B3C.0180.0800.0000.102E  ;2932  ALU.C?
U 102E. 0000.003C.0180.0800.0000.1094  ;2933 =1110 J/PAK1
U 102F. 0800.003C.0180.0A20.0000.120A  ;2934  D_R[4]
U 120A. 0019.0014.0580.0AA0.0000.1094  ;2935  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
            ;2936
            ;2937 ; PACK NORMALIZED EXPECTED QUOTIENT
            ;2938
            ;2939 =0
U 1094. 0000.003D.0180.0800.0000.1147  ;2940 PAK1: CALL[UNSEQA]  ; PACK NORMALIZED EXPECTED PRODUCT
U 1095. 0800.003C.0180.0A30.0000.120B  ;2941  D_R[6]
U 120B. 080D.0034.0180.0A08.0000.120C  ;2942  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 120C. 0001.003C.0180.0AB0.0000.120D  ;2943  R[6]_D
U 120D. 0001.003C.0180.09F0.0000.120E  ;2944  RC[0E]_D  ; SAVE NORMALIZED EXPECTED PRODUCT HI
U 120E. 0800.003C.0180.0A38.0000.120F  ;2945  D_R[7]
U 120F. 0001.003C.0180.09E0.0000.1210  ;2946  RC[0C]_D  ; SAVE NORMALIZED EXPECTED PRODUCT LO
            ;2947
            ;2948 ; PACK OPERANDS FOR MULTIPLY OPERATION
            ;2949
U 1210. 0810.0038.0180.0930.0000.1211  ;2950  D_RC[6]
U 1211. 0001.003C.0180.0AA0.0000.1212  ;2951  R[4]_D   ; RESTORE  HI
U 1212. 0810.0038.0180.0928.0000.1213  ;2952  D_RC[5]
U 1213. 0001.003C.0180.0AA8.0000.1096  ;2953  R[5]_D   ; RESTORE  LO
U 1096. 0000.003D.0180.0800.0000.1147  ;2954 =0 CALL[UNSEQA]
U 1097. 0800.003C.0180.0A30.0000.1214  ;2955  D_R[6]
U 1214. 080D.0034.0180.0A08.0000.1215  ;2956  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 1215. 0003.003C.0180.0AB0.0000.1216  ;2957  R[6]_0   ; MULTIPLICAND HI MASKED
U 1216. 0003.003C.0180.09A0.0000.1217  ;2958  RC[4]_0   ; SAVE
U 1217. 0001.003C.0180.0AC0.0000.1218  ;2959  R[8]_D   ; MULTIPLIER HI
U 1218. 0001.003C.0180.0AB8.0000.1219  ;2960  R[7]_D   ; MULTIPLICAND LO
U 1219. 0001.003C.0180.0998.0000.121A  ;2961  RC[3]_D   ; SAVE
U 121A. 0001.003C.0180.0AC8.0000.1098  ;2962  R[9]_D   ; MULTIPLIER LO
            ;2963
            ;2964 ; LOAD OPERANDS, MULTIPLY.
```

```
            ;2965
U 1098. 0000.003D.0180.0800.0000.1020  ;2966 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
            ;2967
U 1099. 0000.003C.0180.0800.0000.109A  ;2968  NOP
U 109A. 0000.003D.0180.0800.0000.1022  ;2969 =0 CALL[MULF]  ; EXECUTE MULF R,R
U 109B. 081C.2034.0180.0A08.0000.121B  ;2970  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 121B. 0001.003C.0180.09E8.0000.121C  ;2971  RC[0D]_D  ; SAVE NORMALIZED PRODUCT HI
U 121C. 0810.0038.0180.0908.0000.121D  ;2972  D_RC[1]
U 121D. 080D.0034.0180.0A10.0000.121E  ;2973  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 121E. 0001.003C.0180.09C8.0000.121F  ;2974  RC[9]_D  ; SAVE UNNORMALIZED PRODUCT HI
U 121F. 001C.0034.01C0.0A00.0000.1220  ;2975  Q_Q.AND.R[0]  ; MASK OUT ROUND BIT
U 1220. 0001.203C.0180.09D8.0000.1221  ;2976  RC[0B]_Q  ; SAVE MASKED NORMALIZED PRODUCT LO
U 1221. 0000.003C.0180.0900.0000.1222  ;2977  LC_RC[0]
U 1222. 0010.0038.0180.09B8.0000.1223  ;2978  RC[7]_LC  ; SAVE UNNORMALIZED PRODUCT LO
U 1223. 0011.2020.0180.0960.0010.1224  ;2979  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED PRODUCT LO
U 1224. 0000.013C.0180.0800.0000.109C  ;2980  Z?
U 109C. 0000.003D.0180.0800.0000.10E6  ;2981 =0 ERROR1
U 109D. 0010.0038.01C0.0968.0000.1225  ;2982  Q_RC[0D]
U 1225. 0011.2020.0180.0970.0010.1226  ;2983  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED PRODUCT HI
U 1226. 0000.013C.0180.0800.0000.109E  ;2984  Z?
U 109E. 0000.003D.0180.0800.0000.10E6  ;2985 =0 ERROR1
U 109F. 0010.0038.01C0.0940.0000.1227  ;2986  Q_RC[8]
U 1227. 0011.2020.0180.0938.0010.1228  ;2987  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1228. 0000.013C.0180.0800.0000.10A0  ;2988  Z?
U 10A0. 0000.003D.0180.0800.0000.10E6  ;2989 =0 ERROR1
U 10A1. 0010.0038.01C0.0950.0000.1229  ;2990  Q_RC[0A]
U 1229. 0011.2020.0180.0948.0010.122A  ;2991  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 122A. 0000.013C.0180.0800.0000.10A2  ;2992  Z?
U 10A2. 0000.003D.0180.0800.0000.10E6  ;2993 =0 ERROR1
U 10A3. 0810.0038.0180.0910.0000.122B  ;2994  D_RC[2]
U 122B. 0018.0038.2DC0.0800.0000.122C  ;2995  Q_K[.28]
U 122C. 001D.0000.0180.0800.0010.122D  ;2996  ALU_D-Q,CLK.UBCC  ; CHECK BIT COUNTER
U 122D. 0000.013C.0180.0800.0000.10A4  ;2997  Z?  ; 24 BITS PROCESSED?
U 10A4. 0019.0000.0580.0990.0000.1091  ;2998 =0 ALU_D-K[.1],RC[2]_ALU,J/MUL2 ; DECREMENT BIT COUNTER
U 10A5. 0000.003C.0180.0800.0000.10A6  ;2999  NOP
            ;3000
            ;3001
            ;3002
            ;3003
            ;3004
            ;3005
            ;3006
            ;3007
            ;3008
```

```
;3009  .PAGE "TEST 29D EMODF, FLOATING ONE BY 0.1"
;3010 ;*****************************************************************
;3011 ;++
;3012 ;
;3013 ; EMODF, FLOATING ONE BY 0.1
;3014 ;
;3015 ;
;3016 ; TEST DESCRIPTION
;3017 ;
;3018 ; THIS TEST DOES A 24X32 BIT MULTIPLY WITH A FLOATING ONE PATTERN
;3019 ; IN THE MULTIPLICAND HI (MC1) AND MULTIPLICAND EXTENSION (MCX)
;3020 ; REGISTERS, WITH MULTIPLIER HI (MP1) CONTAINING ONLY THE HIDDEN
;3021 ; BIT SET.
;3022 ;
;3023 ;
;3024 ;
;3025 ;
;3026 ;
;3027 ; TEST SEQUENCE   DESCRIPTION
;3028 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;3029 ; 2  PACK THE DATA
;3030 ; 3  PERFORM THE MULTIPLY
;3031 ; 4  VERIFY RESULTS
;3032 ;
;3033 ; FPA UCODE USED
;3034 ;
;3035 ; LOCATION CONTENTS
;3036 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;3037 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;3038 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;3039 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;3040 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;3041 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;3042 ; 028: NAD/028
;3043 ; 0E3: FADC/A,BSC/FAL.HL,OPLD/LDR.R,NAD/0E4
;3044 ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;3045 ; 0EA: FADC/B,BSC/FAL.HL,OPLD/MC1,NAD/0C8
;3046 ; 0CE: OPLD/INIT,NAD/028
;3047 ; 028: NAD/028
;3048 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;3049 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;3050 ; 0CA: NAD/028
;3051 ; 028: NAD/028
;3052 ; 0CC BSC/INTH,FPSYNC,WAIT,NAD/0CD
;3053 ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;3054 ; 0CE: OPLD/INIT,NAD/028
;3055 ;
;3056 ; LOGIC DESCRIPTION
;3057 ;
;3058 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;3059 ; LOGIC ARE ON THE FML AND FMH MODULES.
;3060 ;
;3061 ; ERROR DESCRIPTION
;3062 ;
;3063 ; EXPECTED INTEGER PRODUCT HI
```

```
          ;3064 ; RECEIVED INTEGER PRODUCT HI
          ;3065 ; EXPECTED INTEGER PRODUCT LO
          ;3066 ; RECEIVED INTEGERPRODUCT LO
          ;3067 ; UNPACKED DATA HI
          ;3068 ; UNPACKED DATA LO
          ;3069 ; PACKED MULTIPLIER HI (MP1)
          ;3070 ; PACKED MULTIPLICAND HI (MC1)
          ;3071 ; MULTIPLICAND EXTENSION (MCI)
          ;3072 ; BIT COUNTER
          ;3073 ;
          ;3074 ;
          ;3075 ;
          ;3076 ; SYNC POINT
          ;3077 ;
          ;3078 ;
          ;3079 ;
          ;3080 ;--
          ;3081 ;
          ;3082 ;*************************************************************************
          ;3083
          ;3084
          ;3085
          ;3086 ;/////////////////////////////////////////////////////
          ;3087 ;*
          ;3088 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
          ;3089
          ;3090
          ;3091 =0
U 10A6, 0018,0039,F580,09F8,0000,10D3  ;3092 MULT5: NEWTST[.A]
U 10A7, 0818,0038,4D80,0800,0000,122E  ;3093  D_K[.FF00]
U 122E, 0819,0030,4180,0800,0000,122F  ;3094  D_D.OR.K[.80]
U 122F, 0001,0028,0180,0A88,0000,1230  ;3095  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 1230, 0818,0038,7980,0800,0000,1231  ;3096  D_K[.30]
U 1231, 0819,0030,8980,0800,0000,1232  ;3097  D_D.OR.K[.D]
U 1232, 0001,003C,0180,09A8,0000,10A8  ;3098  RC[5]_D   ; BIT COUNT OF 61
U 10A8, 0000,003D,0180,0800,0000,10E5  ;3099 =0 ERLOOP
U 10A9, 0010,0038,0180,0928,0082,1233  ;3100 MUL5: SC_RC[5]  ; SC GETS PATTERN
U 1233, 0003,003C,0180,0AA8,0000,1234  ;3101  R[5]_0   ; MULTIPLIER LO GETS 0
U 1234, 0003,001C,0180,0AA0,0000,1235  ;3102  R[4]_NOT.MASK  ; FLOATING 1
          ;3103
          ;3104 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
          ;3105
U 1235, 0800,003C,0180,0A20,0000,1236  ;3106  D_R[4]
U 1236, 0001,003C,0180,09D0,0000,1237  ;3107  RC[0A]_D  ; UNPACKED DATA HI
U 1237, 0000,003C,5180,0800,0084,7238  ;3108  SC_K[.1E]
U 1238, 0003,001C,01C0,0800,0000,1239  ;3109  Q_NOT.MASK
U 1239, 001D,0030,0180,09F0,0000,123A  ;3110  RC[0E]_D.OR.Q  ; EXPECTED INTEGER PRODUCT HI
U 123A, 0800,003C,0180,0A28,0000,123B  ;3111  D_R[5]
U 123B, 0001,003C,0180,09E0,0000,123C  ;3112  RC[0C]_D  ; EXPECTED INTEGER PRODUCT LO
U 123C, 0001,003C,0180,09C8,0000,10AA  ;3113  RC[9]_D   ; UNPACKED DATA LO
          ;3114
          ;3115 ; PACK OPERANDS IN R4, R5  FOR MULTIPLY OPERATION
          ;3116
U 10AA, 0000,003D,0180,0800,0000,1147  ;3117 =0 CALL[UNSEQA]
U 10AB, 0800,003C,0180,0A30,0000,123D  ;3118  D_R[6]
```

```
U 123D, 080D.0034.0180.0A08.0000.123E  ;3119  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 123E, 0003.003C.0180.0AC0.0000.123F  ;3120  R[8]_0   ; MULTIPLIER HI MASKED (MP1)
U 123F, 0003.003C.0180.09C0.0000.1240  ;3121  RC[8]_0   ; SAVE
U 1240, 0001.003C.0180.0AB0.0000.1241  ;3122  R[6]_D   ; MULTIPLICAND HI (MC1)
U 1241, 0001.003C.0180.09B8.0000.1242  ;3123  RC[7]_D   ; SAVE
U 1242, 0003.003C.0180.0AC8.0000.1243  ;3124  R[9]_0   ; MULTIPLICAND EXT. (MCI)
U 1243, 0003.003C.0180.09B0.0000.10AC  ;3125  RC[6]_0   ; SAVE
          :3126
          :3127 ; LOAD OPERANDS, MULTIPLY.
          :3128
U 10AC, 0000.003D.0180.0800.0000.1020  ;3129 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
U 10AD, 0000.003C.0180.0800.0000.10AE  ;3130  NOP
U 10AE, 0000.003D.0180.0800.0000.1036  ;3131 =0 CALL[EMODF]  ; EXECUTE EMODF
U 10AF, 0001.203C.0180.09E8.0000.1244  ;3132  RC[0D]_Q  ; SAVE INTEGER PRODUCT HI
U 1244, 0001.003C.0180.09D8.0000.1245  ;3133  RC[0B]_D  ; SAVE INTECER PRODUCT LO
U 1245, 0010.0038.01C0.0960.0000.1246  ;3134  Q_RC[0C]
U 1246, 0011.2020.0180.0958.0010.1247  ;3135  ALU_Q.XOR.RC[0B],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1247, 0000.013C.0180.0800.0000.10B0  ;3136  Z?
U 10B0, 0000.003D.0180.0800.0000.10E6  ;3137 =0 ERROR1
U 10B1, 0010.0038.01C0.0970.0000.1248  ;3138  Q_RC[0E]
U 1248, 0011.2020.0180.0968.0010.1249  ;3139  ALU_Q.XOR.RC[0D],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 1249, 0000.013C.0180.0800.0000.10B2  ;3140  Z?
U 10B2, 0000.003D.0180.0800.0000.10E6  ;3141 =0 ERROR1
U 10B3, 0810.0038.0180.0928.0000.124A  ;3142  D_RC[5]
U 124A, 0018.0038.2DC0.0800.0000.124B  ;3143  Q_K[.28]
U 124B, 001D.0000.0180.0800.0010.124C  ;3144  ALU_D-Q,CLK.UBCC  ; CHECK BIT COUNTER
U 124C, 0000.013C.0180.0800.0000.10B4  ;3145  Z?    ; 24 BITS PROCESSED?
U 10B4, 0019.0000.0580.09A8.0000.10A9  ;3146 =0 ALU_D-K[.1],RC[5]_ALU,J/MUL5 ; DECREMENT BIT COUNTER
U 10B5, 0000.003C.0180.0800.0000.10B6  ;3147  NOP
          :3148
          :3149
          :3150
          :3151
```

```
;3152  .PAGE "TEST 29E EMODF, FLOATING ONE BY 0.1"
;3153 ;*****************************************************************
;3154 ;**
;3155 ;
;3156 ; EMODF, FLOATING ONE BY 0.1
;3157 ;
;3158 ;
;3159 ; TEST DESCRIPTION
;3160 ;
;3161 ; THIS TEST DOES A 24X32 BIT MULTIPLY WITH A FLOATING ONE PATTERN
;3162 ; IN THE MULTIPLIER HI (MP1) WITH THE MULTIPLICAND HI (MC1) AND
;3163 ; MULTIPLICAND EXTENSION (MCX) REGISTERS, CONTAINING ONLY THE
;3164 ; HIDDEN BIT SET.
;3165 ;
;3166 ;
;3167 ;
;3168 ;
;3169 ;
;3170 ; TEST SEQUENCE   DESCRIPTION
;3171 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;3172 ; 2  PACK THE DATA
;3173 ; 3  PERFORM THE MULTIPLY
;3174 ; 4  VERIFY RESULTS
;3175 ;
;3176 ; FPA UCODE USED
;3177 ;
;3178 ; LOCATION CONTENTS
;3179 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;3180 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;3181 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;3182 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;3183 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;3184 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;3185 ; 028: NAD/028
;3186 ; 0E3: FADC/A,BSC/FAL.HL,OPLD/LDR.R,NAD/0E4
;3187 ; 0E2: FADC/B,BSC/FAL.LH,OPLD/MP1,NAD/0C8
;3188 ; 0EA: FADC/B,BSC/FAL.HL,OPLD/MC1,NAD/0C8
;3189 ; 0CE: OPLD/INIT,NAD/028
;3190 ; 028: NAD/028
;3191 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;3192 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;3193 ; 0CA: NAD/028
;3194 ; 028: NAD/028
;3195 ; 0CC BSC/INTH,FPSYNC,WAIT,NAD/0CD
;3196 ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;3197 ; 0CE: OPLD/INIT,NAD/028
;3198 ;
;3199 ; LOGIC DESCRIPTION
;3200 ;
;3201 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;3202 ; LOGIC ARE ON THE FML AND FMH MODULES.
;3203 ;
;3204 ; ERROR DESCRIPTION
;3205 ;
;3206 ; EXPECTED INTEGER PRODUCT HI
```

```
            ;3207 ; RECEIVED INTEGER PRODUCT HI
            ;3208 ; EXPECTED INTEGER PRODUCT LO
            ;3209 ; RECEIVED INTEGERPRODUCT LO
            ;3210 ; UNPACKED DATA HI
            ;3211 ; UNPACKED DATA LO
            ;3212 ; PACKED MULTIPLIER HI (MP1)
            ;3213 ; PACKED MULTIPLICAND HI (MC1)
            ;3214 ; MULTIPLICAND EXTENSION (MCI)
            ;3215 ; BIT COUNTER
            ;3216 ;
            ;3217 ;
            ;3218 ;
            ;3219 ; SYNC POINT
            ;3220 ;
            ;3221 ;
            ;3222 ;
            ;3223 ;--
            ;3224 ;
            ;3225 ;*****************************************************************
            ;3226
            ;3227 ;/////////////////////////////////////////////////
            ;3228 ;*
            ;3229 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
            ;3230
            ;3231
            ;3232 =0
U 10B6, 0018.0039.F580.09F8.0000.10D3  ;3233 MULT6: NEWTST[.A]
U 10B7, 0818.0038.4D80.0800.0000.124D  ;3234  D_K[.FF00]
U 124D, 0819.0030.4180.0800.0000.124E  ;3235  D_D.OR.K[.80]
U 124E, 0001.0028.0180.0A88.0000.124F  ;3236  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 124F, 0818.0038.7980.0800.0000.1250  ;3237  D_K[.30]
U 1250, 0819.0030.8980.0800.0000.1251  ;3238  D_D.OR.K[.D]
U 1251, 0001.003C.0180.09A8.0000.10B8  ;3239  RC[5]_D   ; BIT COUNT OF 61
U 10B8, 0000.003D.0180.0800.0000.10E5  ;3240 =0 ERLOOP
U 10B9, 0010.0038.0180.0928.0082.1252  ;3241 MUL6: SC_RC[5]  ;  SC GETS PATTERN
U 1252, 0003.003C.0180.0AA8.0000.1253  ;3242  R[5]_0   ; MULTIPLIER LO GETS 0
U 1253, 0003.001C.0180.0AA0.0000.1254  ;3243  R[4]_NOT.MASK  : FLOATING 1
            ;3244
            ;3245 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;3246
U 1254, 0800.003C.0180.0A20.0000.1255  ;3247  D_R[4]
U 1255, 0001.003C.0180.09D0.0000.1256  ;3248  RC[0A]_D  ; UNPACKED DATA HI
U 1256, 0000.003C.5180.0800.0084.7257  ;3249  SC_K[.1E]
U 1257, 0003.001C.01C0.0800.0000.1258  ;3250  Q_NOT.MASK
U 1258, 001D.0030.0180.09F0.0000.1259  ;3251  RC[0E]_D.OR.Q  ; EXPECTED INTEGER PRODUCT HI
U 1259, 0800.003C.0180.0A28.0000.125A  ;3252  D_R[5]
U 125A, 0001.003C.0180.09E0.0000.125B  ;3253  RC[0C]_D  ; EXPECTED INTGER PRODUCT LO
U 125B, 0001.003C.0180.09C8.0000.10BA  ;3254  RC[9]_D   ; UNPACKED DATA HI
            ;3255
            ;3256
            ;3257 ; PACK OPERANDS R4, R5  FOR MULTIPLY OPERATION
            ;3258
U 10BA, 0000.003D.0180.0800.0000.1147  ;3259 =0 CALL[UNSEQA]
U 10BB, 0800.003C.0180.0A30.0000.125C  ;3260  D_R[6]
U 125C, 080D.0034.0180.0A08.0000.125D  ;3261  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
```

```
U 125D. 0001.003C.0180.0AC0.0000.125E  ;3262  R[8]_D   ; MULTIPLIER HI MASKED (MP1)
U 125E. 0001.003C.0180.09C0.0000.125F  ;3263  RC[8]_D   ; SAVE
U 125F. 0003.003C.0180.0AB0.0000.1260  ;3264  R[6]_0   ; MULTIPLICAND HI (MC1)
U 1260. 0003.003C.0180.09B8.0000.1261  ;3265  RC[7]_0   ; SAVE
U 1261. 0003.003C.0180.0AC8.0000.1262  ;3266  R[9]_0   ; MULTIPLICAND EXT. (MC1)
U 1262. 0003.003C.0180.09B0.0000.10BC  ;3267  RC[6]_0   ; SAVE
          ;3268
          ;3269 ; LOAD OPERANDS, MULTIPLY.
          ;3270
U 10BC. 0000.003D.0180.0800.0000.1020  ;3271 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
U 10BD. 0000.003C.0180.0800.0000.10BE  ;3272  NOP
U 10BE. 0000.003D.0180.0800.0000.1036  ;3273 =0 CALL[EMODF]  ; EXECUTE EMODF
U 10BF. 0001.203C.0180.09E8.0000.1263  ;3274  RC[0D]_Q  ; SAVE INTEGER PRODUCT HI
U 1263. 0001.003C.0180.09D8.0000.1264  ;3275  RC[0B]_D  ; SAVE INTEGER PRODUCT LO
U 1264. 0010.0038.01C0.0960.0000.1265  ;3276  Q_RC[0C]
U 1265. 0011.2020.0180.0958.0010.1266  ;3277  ALU_Q.XOR.RC[0B],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1266. 0000.013C.0180.0800.0000.10C0  ;3278  Z?
U 10C0. 0000.003D.0180.0800.0000.10E6  ;3279 =0 ERROR1
U 10C1. 0010.0038.01C0.0970.0000.1267  ;3280  Q_RC[0E]
U 1267. 0011.2020.0180.0968.0010.1268  ;3281  ALU_Q.XOR.RC[0D],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 1268. 0000.013C.0180.0800.0000.10C2  ;3282  Z?
U 10C2. 0000.003D.0180.0800.0000.10E6  ;3283 =0 ERROR1
U 10C3. 0810.0038.0180.0928.0000.1269  ;3284  D_RC[5]
U 1269. 0018.0038.2DC0.0800.0000.126A  ;3285  Q_K[.28]
U 126A. 001D.0000.0180.0800.0010.126B  ;3286  ALU_D-Q,CLK.UBCC  ; CHECK BIT COUNTER
U 126B. 0000.013C.0180.0800.0000.10C4  ;3287  Z?    ; 24 BITS PROCESSED?
U 10C4. 0019.0000.0580.09A8.0000.10B9  ;3288 =0 ALU_D-K[.1],RC[5]_ALU,J/MUL6 ; DECREMENT BIT COUNTER
U 10C5. 0000.003C.0180.0800.0000.10C6  ;3289  NOP
          ;3290
          ;3291
          ;3292 ; OPCODE DATA
          ;3293 ;* 8
          ;3294 ; MULF R,R
          ;3295 ;$ 5544, 0056
          ;3296 ; MULL R,R
          ;3297 ;$ 55C4, 0056
          ;3298 ; MULD R,R
          ;3299 ;$ 5564, 0056
          ;3300 ; EMODF
          ;3301 ;$5554, 0056
          ;3302 ; EMODD
          ;3303 ;$ 5574, 0056
          ;3304
```

```
            ;3305  .PAGE "TEST 29F RESERVED"
            ;3306 =0
U 10C6, 0000,003D,0180,0800,0000,10D3  ;3307 TSTXX1: NEWTST
U 10C7, 0000,003C,0180,0800,0000,10C8  ;3308  NOP
            ;3309
```

```
            ;3310  .PAGE "TEST 2A0 RESERVED"
            ;3311 =0
U 10C8, 0000,003D,0180,0800,0000,10D3  ;3312 TSTXX2: NEWTST
U 10C9, 0000,003C,0180,0800,0000,10CA  ;3313  NOP
            ;3314
```

```
            ;3315  .PAGE "DUMMY NEWTST"
            ;3316 =0
U 10CA, 0000,003D,0180,0800,0000,10D3  ;3317 TSTXX3: NEWTST
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2821
ESKAR65.MCR                    MICRO2  1P(00)    26-JUL-85  17:10:43     Page   8
;                             Location / Line Number Index

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 2474= 2475=
U 1008  1918= 1919= 1875 2007= 2018= 2019= 1876 2008=
U 1010  2049= 2050= 2056= 2057= 2060= 2061= 2482= 2483=
U 1018  2064= 2065= 2068= 2069= 2072= 2073= 2733= 2734=
U 1020  2110= 2111= 2133= 2134= 2135= 2136= 2598= 2599=
U 1028  2152= 2155= 2156= 2157= 2158= 2159= 2933= 2934=
U 1030  2160= 2161= 2170= 2171= 2173= 2174= 2190= 2191=
U 1038  2192= 2193= 2194= 2195= 2196= 2197= 2198= 2199=
U 1040  2212= 2213= 2237= 2238= 2247= 2248= 2310= 2311=
U 1048  2313= 2314= 2315= 2316= 2318= 2319= 1939: 1877
U 1050  2404= 2408= 2409= 2410= 2415= 2416= 2417= 2418=
U 1058  2425= 2426= 1942: 1878 2429= 2430= 1945: 1879
U 1060  2452= 2453= 2454= 2455= 2462= 2463= 2466= 2467=
U 1068  2469= 2470= 2560= 2561= 2573= 2574= 2580= 2581=
U 1070  2582= 2583= 2590= 2591= 2594= 2595= 2683= 2684=
U 1078  2696= 2697= 2716= 2717= 2740= 2741= 2754= 2755=
U 1080  2766= 2767= 2768= 2769= 2780= 2781= 2784= 2785=
U 1088  2788= 2789= 2792= 2793= 2797= 2798= 2883= 2884=
U 1090  2896= 2897= 2916= 2917= 2940= 2941= 2954= 2955=
U 1098  2966= 2968= 2969= 2970= 2981= 2982= 2985= 2986=
U 10A0  2989= 2990= 2993= 2994= 2998= 2999= 3092= 3093=
U 10A8  3099= 3100= 3117= 3118= 3129= 3130= 3131= 3132=
U 10B0  3137= 3138= 3141= 3142= 3146= 3147= 3233= 3234=
U 10B8  3240= 3241= 3259= 3260= 3271= 3272= 3273= 3274=
U 10C0  3279= 3280= 3283= 3284= 3288= 3289= 3307= 3308=
U 10C8  3312= 3313= 3317= 1880 1900 1901 1902 1903
U 10D0  1904 1905 1906 1916 1917 1920 1921 1922
U 10D8  1923 1924 1925 1926 1927 1928 1929 1930
U 10E0  1931 1932 1933 1934 1935 1936 1937 1938
U 10E8  1940 1941 1973 1974 1980 1981 1982 2003
U 10F0  2004 2005 2006 2016 2017 2020 2027 2028
U 10F8  2031 2032 2034 2035 2037 2038 2040 2041
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 2042 2044 2045 1870: 1871: 1872: 1873:
U 1110  2046 2052 2053 2076 2077 2078 2082 2083
U 1118  2084 2093 2094 2095 2096 2097 2098 2099
U 1120  2112 2113 2114 2115 2116 2117 2118 2119
U 1128  2120 2121 2122 2123 2137 2138 2139 2140
U 1130  2141 2142 2172 2175 2176 2177 2218 2219
U 1138  2220 2221 2223 2224 2240 2242 2244 2249
U 1140  2250 2251 2252 2253 2254 2255 2256 2265
U 1148  2266 2267 2268 2269 2270 2271 2272 2273
U 1150  2274 2275 2276 2277 2278 1965: 2285 2286
U 1158  2287 2288 2289 2290 2291 2292 2293 2294
U 1160  2308 2309 2312 2317 2320 2327 2328 2329
U 1168  2411 2412 2413 2414 2419 2423 2424 2427
U 1170  2428 2434 2437 2438 2439 2440 2441 2442
U 1178  2443 2444 2445 2446 2447 2448 2456 2460
U 1180  2461 2464 2465 2468 2471 2472 2473 2476
U 1188  2477 2478 2479 2480 2481 2562 2563 2564
U 1190  2565 2566 2567 2568 2569 2570 2571 2572
U 1198  2575 2584 2588 2589 2592 2593 2596 2597
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2822
ESKAR65.MCR                    MICRO2  1P(00)    26-JUL-85  17:10:43     Page   8

;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2685 2686 2687 2688 2689 2690 2691 2692
U 11A8  2693 2694 2695 2698 2699 2703 2704 2705
U 11B0  2706 2707 2708 2709 2710 2711 2712 2718
U 11B8  2719 2720 2721 2722 2723 2727 2728 2729
U 11C0  2730 2731 2732 2735 2742 2743 2744 2745
U 11C8  2746 2750 2751 2752 2753 2756 2757 2758
U 11D0  2759 2760 2761 2762 2770 2771 2772 2773
U 11D8  2774 2775 2776 2777 2778 2779 2782 2783
U 11E0  2786 2787 2790 2791 2794 2795 2796 2885
U 11E8  2886 2887 2888 2889 2890 2891 2892 2893
U 11F0  2894 2895 2898 2899 2903 2904 2905 2906
U 11F8  2907 2908 2909 2910 2911 2912 2918 2919
U 1200  2920 2921 2922 2923 2927 2928 2929 2930
U 1208  2931 2932 2935 2942 2943 2944 2945 2946
U 1210  2950 2951 2952 2953 2956 2957 2958 2959
U 1218  2960 2961 2962 2971 2972 2973 2974 2975
U 1220  2976 2977 2978 2979 2980 2983 2984 2987
U 1228  2988 2991 2992 2995 2996 2997 3094 3095
U 1230  3096 3097 3098 3101 3102 3106 3107 3108
U 1238  3109 3110 3111 3112 3113 3119 3120 3121
U 1240  3122 3123 3124 3125 3133 3134 3135 3136
U 1248  3139 3140 3143 3144 3145 3235 3236 3237
U 1250  3238 3239 3242 3243 3247 3248 3249 3250
U 1258  3251 3252 3253 3254 3261 3262 3263 3264
U 1260  3265 3266 3267 3275 3276 3277 3278 3281
U 1268  3282 3285 3286 3287
U 1270 - 12A7 Unused
U 12A8    1966:
U 12B0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2823
ESKAR65.MCR                    MICRO2  1P(00)    26-JUL-85  17:10:43     Page   8

           Words not
           in bounds
 ESK00DEF          0
 ESK00MAC          0
 ESKARMAC          0
  ESKAR65        637

Used               637
Remaining

Total microwords used in memory U: 637
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                                 SEQ 2824
ESKAR65.MCR                    MICRO2  1P(00)    26-JUL-85  17:10:43           Page   8


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR65.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```

ZZ-ESKAR-V22 Table of Contents
   ESKAR66.MCR   MICRO2 1P(00)   26-JUL-85 17 SEQ 2825

; 3 Control ROM Field Definitions
; 548 Register Transfer Macro Definitions
; 1413 Non-transfer Functions
; 1485 Branch Enable Macro Definitions
; 1564 MICRO DIAGNOSTIC DEFINED MACROS
; 1807 MODULE REVISION HISTORY
; 1843 MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES
; 2333 TEST 2A1 EMODD FLOATING ONE BY 0.1
; 2500 TEST 2A2 EMODD 0.1 BY FLOATING ONE
; 2652 TEST 2A3 MULTIPLY DOUBLE FLOATING ONE BY 0.1
; 2858 TEST 2A4 MULTIPLY DOUBLE, 0.1 BY FLOATING ONE
; 3071 TEST 2A5 VERIFY THE MULTIPLIER ROMS
; 3384 TEST 2A6 RESERVED
; 3389 DUMMY NEWTST

```
;1  .NOLIST
;1804  .LIST
;1805
```

```
        ;1806 .REGION/1000,13FF
```

```
;1807  .PAGE "MODULE REVISION HISTORY"
;1808 ;*****************************************************************************
;1809 ;
;1810 ;
;1811 ; MODULE NAME: ESKAR66
;1812 ;
;1813 ; CREATION DATE:  SEPTEMBER 1982
;1814 ; AUTHOR:  DAN GRIGORE
;1815 ;
;1816 ; PLEASE INCLUDE DETAILED DESCRIPTION ON THE FOLLOWING ITEMS:
;1817 ;
;1818 ; -  TEST NUMBER ON WHICH CHANGES WERE MADE
;1819 ;
;1820 ; -  WHAT CHANGE WAS MADE
;1821 ;
;1822 ; -  WHY WAS THE CHANGE MADE (INCLUDE DATA ON THE AIDS REPORT
;1823 ;    THAT PROMPTED THE CHANGE IF APLICABLE)
;1824 ;
;1825 ; -  EDIT NUMBER (SEE ESKAR3C FOR MORE DETAILS ON THIS)
;1826 ;    NOTE:  THE EDIT NUMBER IN THIS MODULE HEADER MUST
;1827 ;           MATCH  THE  EDIT  NUMBER FOR THE CHANGE IN
;1828 ;    ESKAR3C).   MAKE SURE  THAT WHEN  MAKING A
;1829 ;    CHANGE  ALL  THE  INFORMATION  REQUIRED IS
;1830 ;    INCLUDED BOTH IN  THE MODULE HEADER AND IN
;1831 ;    ESKAR3C.
;1832 ;
;1833 ; START OF REVISION HISTORY:
;1834 ;
;1835 ;
;1836 ;
;1837 ;
;1838 ;
;1839 ;*****************************************************************************
;1840
```

```
            ;1841   .PAGE
            ;1842
            ;1843   .TOC "MICRO TRAP HANDLER AND LSI-MONITER CALL ROUTINES"
            ;1844 ;*
            ;1845 ; FOLLOWING ARE THE MICRO TRAP VECTOR ASSIGNMENTS. EACH VECTOR LOADS THE SC
            ;1846 ; WITH A NUMBER, AND LOADS THE D REGISTER WITH THE CONTENTS OF R[0F].
            ;1847 ; R[0F] IS SETUP BY THE PROGRAM TO INDICATE WHETHER A MICRO TRAP IS EXPECTED
            ;1848 ; THIS IS DONE BY SETTING THE BIT IN R[0F] THAT CORRESPONDS TO THE NUMBER
            ;1849 ; THAT IS LOADED INTO THE SC BY THAT PARTICULAR MICRO TRAP.
            ;1850 ;
            ;1851 ; FOR EXAMPLE: IF A TEST EXPECTED A "TB MISS" MICRO TRAP IT WOULD SET BIT
            ;1852 ; 6 IN SCRATCH PAD R[0F]. WHEN THE TRAP OCCURRED, THE TRAP CATCHER
            ;1853 ; ROUTINE WOULD SENSE THAT THE TRAP WAS EXPECTED AND CLEAR BIT 6 IN
            ;1854 ; R[0F] AND DO A RETURN0.
            ;1855 ;
            ;1856 ; IF THE BIT FOR THE PARTICULAR TRAP IS NOT SET IN R[0F], THE TRAP ROUTINE
            ;1857 ; READS THE ERROR REGISTERS, SAVES THEM FOR TYPEOUT, AND MAKES AN ERROR CALL
            ;1858 ; TO THE CONSOLE.
            ;1859 ;
            ;1860 ;-
U 1100, 0000.003C.1980.0800.0084.7003  ;1861 1100: SC_K[ZERO],J/TRPH1 ; SYSTEM INIT
U 1101, 0000.003C.0580.0800.0084.7003  ;1862 1101: SC_K[.1],J/TRPH1 ; DATA UNALIGNED
U 1102, 0000.003C.0980.0800.0084.7003  ;1863 1102: SC_K[.2],J/TRPH1 ; X PAGE ERROR
U 1103, 0000.003C.0D80.0800.0084.7003  ;1864 1103: SC_K[.3],J/TRPH1 ; TB MODIFY
U 1104, 0000.003C.1180.0800.0084.7003  ;1865 1104: SC_K[.4],J/TRPH1 ; TB PROT ERROR
U 1105, 0000.003C.D580.0800.0084.7003  ;1866 1105: SC_K[.6],J/TRPH1 ; TB MISS
U 1106, 0000.003C.D980.0800.0084.7003  ;1867 1106: SC_K[.9],J/TRPH1 ; RES FLOAT OPER
U 1107, 0000.003C.5D80.0800.0084.7003  ;1868 1107: SC_K[.7],J/TRPH1 ; TB PARITY
U 1108, 0000.003C.0180.0800.0084.7003  ;1869 1108: SC_K[.8],J/TRPH1 ; CACHE PARITY
U 110C, 0000.003C.8580.0800.0084.7003  ;1870 110C: SC_K[.C],J/TRPH1 ; RDS
U 110D, 0000.003C.8980.0800.0084.7003  ;1871 110D: SC_K[.D],J/TRPH1 ; TIME OUT
U 110E, 0000.003C.6580.0800.0084.7003  ;1872 110E: SC_K[.10],J/TRPH1 ; ODD ADDRESS
U 110F, 0000.003C.6180.0800.0084.7003  ;1873 110F: SC_K[.F],J/TRPH1 ; CS PARITY
U 1001, 0000.003C.81F0.2C00.0000.100A  ;1874 TRPH0: Q_ID[USTACK]
U 100A, 0C00.003C.01E0.0800.0000.100E  ;1875  Q_D,D_Q
U 100E, 0000.003C.8180.3C00.0000.1016  ;1876  ID[USTACK]_D
U 1016, 0C00.003C.01E0.0800.0000.101E  ;1877  Q_D,D_Q
U 101E, 0000.003C.01C0.0A78.0000.1026  ;1878  Q_R[0F]
U 1026, 0001.2024.0180.0800.0010.102E  ;1879  ALU_Q.ANDNOT.MASK,CLK.UBCC ; WAS TRAP EXPECTED?
U 102E, 0000.013C.0180.0800.0000.1002  ;1880  Z?     ; BRANCH IF NO
U 1002, 0001.2036.0180.0AF8.0000.0000  ;1881 =0 ALU_Q.AND.MASK,R[0F]_ALU,RETURN0 ; CLEAR THE BIT AND RETURN
            ;1882 ;*
            ;1883 ; THE FOLLOWING ROUTINE IS ENTERED WHEN A MICRO TRAP WAS NOT EXPECTED.
            ;1884 ; IT PUTS THE CONTENTS OF THE ERROR REGISTERS INTO THE RC SCRATCH PADS
            ;1885 ; FOR TYPEOUT. FOLLOWING IS THE FORMAT OF THE TYPEOUT:
            ;1886 ;
            ;1887 ; DATA: TOP OF THE MICRO STACK(MICRO TRAP ADDRESS)
            ;1888 ;       TRAP CODE--INDICATES WHICH TRAP OCCURRED
            ;1889 ;       FAULT STATUS REGISTER
            ;1890 ;       SBI ERROR REGISTER
            ;1891 ;       TIMEOUT ADDRESS REGISTER
            ;1892 ;       MAINTENANCE REGISTER
            ;1893 ;       CACHE PARITY REGISTER
            ;1894 ;       TB ERROR REGISTER 1
            ;1895 ;       TB ERROR REGISTER 0
```

```
                ;1896 ;-
U 1003, 0018.0038.1D80.09E8.0000.1004  ;1897 TRPH1: RC[0D]_SC
U 1004, 0000.003D.D980.0800.0084.70D5  ;1898 =0 SETRCF[.9]
U 1005, 0000.003C.49F0.2C00.0000.104F  ;1899  Q_ID[TBER0]
U 104F, 0001.203C.4DF0.2DB0.0000.105B  ;1900  RC[6]_Q,Q_ID[TBER1]
U 105B, 0001.203C.65F0.2DB8.0000.105F  ;1901  RC[7]_Q,Q_ID[SBI.ERR]
U 105F, 0001.203C.6DF0.2DD8.0000.10B8  ;1902  RC[0B]_Q,Q_ID[FAULT]
U 10B8, 0001.203C.75F0.2DE0.0000.10B9  ;1903  RC[0C]_Q,Q_ID[MAINT]
U 10B9, 0001.203C.79F0.2DC8.0000.10BA  ;1904  RC[9]_Q,Q_ID[PARITY]
U 10BA, 0001.203C.69F0.2DC0.0000.10BB  ;1905  RC[8]_Q,Q_ID[TIME.ADDR]
U 10BB, 0001.203C.81F0.2DD0.0000.1000  ;1906  RC[0A]_Q,Q_ID[USTACK]
U 1000, 0001.203D.0180.09F0.0000.10CF  ;1907 1000: RC[0E]_Q,ERROR1
                ;1908 ;*
                ;1909 ; THESE ROUTINES ARE USED TO CALL THE CONSOLE WHEN THE
                ;1910 ; MICRO TEST EXECUTES ONE OF THE INSTRUCTIONS:
                ;1911 ;  NEWTST
                ;1912 ;  ERLOOP
                ;1913 ;  ERROR1
                ;1914 ;  ERROR2
                ;1915 ;
U 10BC, 0000.003C.8580.0800.0084.70BD  ;1916 SCOPE: SC_K[.C]
U 10BD, 0803.001C.0180.0800.0000.1008  ;1917  ALU_NOT.MASK,D_ALU
U 1008, 0000.003D.7580.3C00.0000.1145  ;1918 =0 ID[MAINT]_D,CALL,J/FPINIT
U 1009, 0000.003C.65F8.0800.0084.70BE  ;1919  SC_K[.10],Q_0    ; SETUP TO WRITE IPL OF 1F
U 10BE, 0818.0038.8D80.0800.0000.10BF  ;1920  D_K[.1F]    ; ...
U 10BF, 0D00.003C.0180.0800.0000.10C0  ;1921  D_DAL.SC
U 10C0, 0000.003C.3D80.3C00.0000.10C1  ;1922  ID[PSL]_D    ; WRITE THE PSL
U 10C1, 0818.0038.0580.0800.0000.10C2  ;1923  D_K[.1]    ; SETUP TO CLEAR THE CES REGISTER
U 10C2, 0D00.003C.0180.0800.0000.10C3  ;1924  D_DAL.SC
U 10C3, 0F00.003C.3180.3C00.0000.10C4  ;1925  ID[CES]_D,D_0  ; CLEAR THE CES REGISTER
U 10C4, 0000.003C.0180.0800.0000.10C5  ;1926  NOP
U 10C5, 0000.003C.3980.3C00.0000.10C6  ;1927  ID[SIR]_D    ; CLEAR THE SIR REGISTER
U 10C6, 0000.003C.2980.3C00.0000.10C7  ;1928  ID[CLK.CS]_D    ; STOP INTERVAL TIMER
U 10C7, 0000.003C.4980.3C00.0000.10C8  ;1929  ID[TBER0]_D    ; SHUT OFF THE TB
U 10C8, 0818.0038.C180.0800.0000.10C9  ;1930  D_K[.FFFF]
U 10C9, 0000.003C.6580.3C00.0000.10CA  ;1931  ID[SBI.ERR]_D
U 10CA, 0F00.003C.6D80.3C00.0000.10CB  ;1932  ID[FAULT]_D,D_0
U 10CB, 0000.003C.6D80.3C00.0000.10CC  ;1933  ID[FAULT]_D
U 10CC, 0000.003C.6580.3C00.0000.10CD  ;1934  ID[SBI.ERR]_D
U 10CD, 0003.003C.0180.0AF8.0000.105E  ;1935  R[0F]_0,J/CALLCON  ; CALL THE CONSOLE
U 10CE, 0818.0038.0980.0800.0000.105E  ;1936 ERLOOP: D_K[.2],J/CALLCON  ; ERRLOOP CALL
U 10CF, 0810.0038.0180.0978.0000.10D0  ;1937 ERROR1: D_RC[0F]
U 10D0, 0819.0034.4D80.0800.0000.104E  ;1938  D_D.AND.K[.FF00]
U 104E, 0819.0030.1180.0800.0000.105E  ;1939 104E: D_D.OR.K[.4],J/CALLCON
U 10D1, 0810.0038.0180.0978.0000.10D2  ;1940 ERROR2: D_RC[0F]
U 10D2, 0819.0034.4D80.0800.0000.105A  ;1941  D_D.AND.K[.FF00]
U 105A, 0819.0030.D580.0800.0000.105E  ;1942 105A: D_D.OR.K[.6],J/CALLCON
                ;1943 105E:
                ;1944 CALLCON:
U 105E, 0000.003C.1D80.3C00.0000.105E  ;1945  ID[TXDB]_D,J/CALLCON  ; CALL THE CONSOLE
                ;1946 ;*
                ;1947 ; THESE ARE CS SCRATCH LOCATIONS USED BY THE CONSOLE MONITOR PROGRAM
                ;1948 ;-
U 13F0, 0000.003C.0180.0800.0000.13F1  ;1949 13F0: NOP
U 13F1, 0000.003C.0180.0800.0000.13F2  ;1950 13F1: NOP
```

```
U 13F2, 0000,003C,0180,0800,0000,13F3  ;1951 13F2: NOP
U 13F3, 0000,003C,0180,0800,0000,13F4  ;1952 13F3: NOP
U 13F4, 0000,003C,0180,0800,0000,13F5  ;1953 13F4: NOP
U 13F5, 0000,003C,0180,0800,0000,13F6  ;1954 13F5: NOP
U 13F6, 0000,003C,0180,0800,0000,13F7  ;1955 13F6: NOP
U 13F7, 0000,003C,0180,0800,0000,13F8  ;1956 13F7: NOP
U 13F8, 0000,003C,0180,0800,0000,13F9  ;1957 13F8: NOP
U 13F9, 0000,003C,0180,0800,0000,13FA  ;1958 13F9: NOP
U 13FA, 0000,003C,0180,0800,0000,13FB  ;1959 13FA: NOP
U 13FB, 0000,003C,0180,0800,0000,13FC  ;1960 13FB: NOP
U 13FC, 0000,003C,0180,0800,0000,13FD  ;1961 13FC: NOP
U 13FD, 0000,003C,0180,0800,0000,13FE  ;1962 13FD: NOP
U 13FE, 0000,003C,0180,0800,0000,13FF  ;1963 13FE: NOP
U 13FF, 0000,003C,0180,0800,0000,1155  ;1964 13FF: NOP
U 1155, 0001,203E,59F0,2DE8,0000,0002  ;1965 1155: RC[0D]_Q,Q_ID[ACC.2],RETURN2
U 12AA, 0001,203E,59F0,2DE8,0000,0002  ;1966 12AA: RC[0D]_Q,Q_ID[ACC.2],RETURN2
            ;1967 ;*
            ;1968 ; THIS SUBROUTINE IS USED TO INCREMENT THE SUBTEST NUMBER
            ;1969 ; THE SUBTEST NUMBER IS STORED IN BYTE 1 OF REGISTER RC[0F] AS A
            ;1970 ; TWO'S COMPLIMENT NUMBER. THE CONSOLE PROGRAM NEGATES THIS NUMBER BEFORE
            ;1971 ; TYPING IT.
            ;1972 ;-
U 10D3, 0810,0038,4D80,0978,0000,10D4  ;1973 SUBTST: D_RC[0F],KMX/.FF00
U 10D4, 0019,4016,4D80,09F8,0000,0001  ;1974  RC[0F]_D+K[.FF00],WORD,RETURN1
            ;1975 ;*
            ;1976 ; THIS ROUTINE IS USED TO SET THE NUMBER OF DATA WORDS TO TYPE INTO THE
            ;1977 ; LOW BYTE OF RC[0F] WITHOUT CHANGING THE OTHER BYTES. IT MUST BE CALLED
            ;1978 ; WITH THE NUMBER TO PUT IN THE LOWER BYTE IN THE SC.
            ;1979 ;-
U 10D5, 0010,0038,01C0,0978,0000,10D6  ;1980 SETRCF: Q_RC[0F]
U 10D6, 0019,2034,4DC0,0800,0000,10D7  ;1981  Q_Q.AND.K[.FF00]
U 10D7, 0019,2032,1D80,09F8,0000,0001  ;1982  RC[0F]_Q.OR.SC,RETURN1
            ;1983
            ;1984 ;*********************************************************
            ;1985 ;
            ;1986 ; MAIN MACHINE/FPA SUBROUTINES
            ;1987 ;
            ;1988 ;*********************************************************
            ;1989
            ;1990 ; SHIFT D REGISTTER CONTENTS
            ;1991
            ;1992
            ;1993 ; STATE COUNTER MUST BE INITIALIZED TO REQUIRED SHIFT COUNT.
            ;1994
            ;1995
U 10D8, 0000,003C,01F8,0800,0000,10D9  ;1996 SHIFTD: Q_0
U 10D9, 0500,003C,0180,0800,0000,10DA  ;1997  D_D.LEFT  ;
U 10DA, 0000,003C,0580,0800,1414,B0DB  ;1998  EALU_STATE-K[.1],CLK.UBCC ;
U 10DB, 0000,123C,0180,0800,0000,100B  ;1999  EALU.Z?
U 100B, 0000,003C,0180,0800,0000,10D8  ;2000 =1011 J/SHIFTD  ;
U 100F, 0000,003E,0180,0800,0000,0001  ;2001  RETURN1
            ;2002
            ;2003 ;*
            ;2004 ; THIS ROUTINE TAKES THE TRAP ADDRESS PREVIOUSLY LOADED IN THE D REGISTER
            ;2005 ; OR'S IN BIT 15, SHIFTS AND LOADS THE FPA STATUS REGISTER FOR A TRAP
```

```
            ;2006 ; TO THAT ADDRESS.
            ;2007 ;-
            ;2008
U 10DC, 0819,0030,4580,0800,0000,10DD  ;2009 TRAP1: D_D.OR.K[.8000]
U 10DD, 0000,003C,6580,0800,1404,700C  ;2010  STATE_K[.10]
U 100C, 0000,003D,0180,0800,0000,10D8  ;2011 =0 CALL[SHIFTD]
U 100D, 0000,003C,5980,3C00,0000,10DE  ;2012  ID[ACC.2]_D
U 10DE, 0000,00FE,0380,0800,0000,0001  ;2013  TRAP.FPA,RETURN1
            ;2014
            ;2015 ;*
            ;2016 ; THE FOLLOWING ROUTINES GENERATE FPA UPC ADDRESS, AND
            ;2017 ; JUMP TO THE TRAP ROUTINE.
            ;2018 ;-
            ;2019
U 10DF, 0818,0038,A580,0800,0000,10E0  ;2020 TRP.68: D_K[.60]
U 10E0, 0819,0030,0180,0800,0000,10DC  ;2021  D_D.OR.K[.8],J/TRAP1
            ;2022
U 10E1, 0818,0038,D180,0800,0000,10E2  ;2023 TRP.C3: D_K[.C0]
U 10E2, 0819,0030,0D80,0800,0000,10DC  ;2024  D_D.OR.K[.3],J/TRAP1
            ;2025
            ;2026
U 10E3, 0818,0038,D180,0800,0000,10E4  ;2027 TRP.C8: D_K[.C0]
U 10E4, 0819,0030,0180,0800,0000,10DC  ;2028  D_D.OR.K[.8],J/TRAP1
            ;2029
U 10E5, 0818,0038,5D80,0800,0000,10E6  ;2030 TRP.CE: D_K[.7]
U 10E6, 0500,003C,0180,0800,0000,10E7  ;2031  D_D.LEFT
U 10E7, 0819,0030,D180,0800,0000,10DC  ;2032  D_D.OR.K[.C0],J/TRAP1
            ;2033
U 10E8, 0818,0038,D180,0800,0000,10E9  ;2034 TRP.CC: D_K[.C0]
U 10E9, 0819,0030,8580,0800,0000,10DC  ;2035  D_D.OR.K[.C],J/TRAP1
            ;2036
U 10EA, 0818,0038,8980,0800,0000,10EB  ;2037 D0: D_K[.D]
U 10EB, 0100,003C,0180,0800,0000,10EC  ;2038  D_D.LEFT2
U 10EC, 0100,003E,0180,0800,0000,0001  ;2039  D_D.LEFT2,RETURN1
            ;2040
            ;2041 =0
U 1010, 0000,003D,0180,0800,0000,10EA  ;2042 TRP.D0: CALL[D0]
U 1011, 0000,003C,0180,0800,0000,10DC  ;2043  J/TRAP1
            ;2044
U 10ED, 0818,0038,CD80,0800,0000,10EE  ;2045 E0: D_K[.F0]
U 10EE, 0819,0026,6580,0800,0000,0001  ;2046  D_D[ANDNOT]K[.10],RETURN1
            ;2047
            ;2048 =0
U 1012, 0000,003D,0180,0800,0000,10ED  ;2049 TRP.E6: CALL[E0]
U 1013, 0819,0030,D580,0800,0000,10DC  ;2050  D_D.OR.K[.6],J/TRAP1
            ;2051
U 10EF, 0818,0038,CD80,0800,0000,10F0  ;2052 TRP.FB: D_K[.F0]
U 10F0, 0819,0030,F580,0800,0000,10F1  ;2053  D_D.OR.K[.A]
U 10F1, 0819,0014,0580,0800,0000,10DC  ;2054  D_D+K[.1],J/TRAP1
            ;2055
            ;2056 ;*
            ;2057 ; THIS ROUTINE STARTS THE INSTRUCTION BUFFER AND WAITS FOR THREE
            ;2058 ; CACHE REFERANCES TO BE GIVEN TO THE IB. THIS GUARANTEES THAT THE
            ;2059 ; IB IS COMPLETELY FULL OF DATA.
            ;2060 ;
```

```
            ;2061 ; BEFORE EXITING THE ROUTINE, THE INSTRUCTION BUFFER IS SHUT OFF.
            ;2062 ;-
U 10F2, 3000,003C,0180,6000,0000,10F3  ;2063 IBLOAD: LOAD.IB,IBC/START
U 10F3, 0000,003C,0180,F800,0000,10F4  ;2064 SYNC02: MCT/ALLOW.IB.READ
U 10F4, 0000,003C,0180,F800,0000,10F5  ;2065  MCT/ALLOW.IB.READ
U 10F5, 0000,003C,0180,F800,0000,10F6  ;2066  MCT/ALLOW.IB.READ
U 10F6, 0000,003C,0180,F800,0000,10F7  ;2067  MCT/ALLOW.IB.READ
U 10F7, 0000,003C,0180,F800,0000,10F8  ;2068  MCT/ALLOW.IB.READ
U 10F8, 1000,003E,0180,F800,0000,0001  ;2069  MCT/ALLOW.IB.READ,IBC/STOP,RETURN1
            ;2070
            ;2071 ;*
            ;2072 ; THIS ROUTINE LOADS THE FAD AR AND BR LATCHES WITH MULTIPLIER AND
            ;2073 ; MULTIPLICAND OPERANDS. R6,R7,R8, AND R9 MUST BE LOADED IN THE
            ;2074 ; CALLING PROGRAM. THE LA EXPONENT LATCH IS LOADED SO THAT WHEN THE
            ;2075 ; NORMALIZED FRACTION PRODUCTS ARE READ BACK ON BUSA, THE NORMALIZER
            ;2076 ; MUX WILL PASS THE NORMALIZED PRODUCT, AND NOT FORCE ZEROS ON THE BUS.
            ;2077 ;-
            ;2078
            ;2079 =0
U 1014, 0000,003D,0180,0800,0000,10DF  ;2080 LD.MUL: CALL[TRP.68]
U 1015, 0018,0038,4180,0AF0,0000,10F9  ;2081  R[0E]_K[,80]
U 10F9, 0000,003C,0180,0A70,0000,10FA  ;2082  LAB_R[0E]  ; FPA AT 068.
U 10FA, 0000,007C,0180,0A70,0000,10FB  ;2083  LAB_R[0E],CPSYNC ; LOAD LA
U 10FB, 0000,007C,0180,0800,0000,10FC  ;2084  CPSYNC   ; FPA AT 069.
U 10FC, 0000,003C,0180,0A30,0000,10FD  ;2085  LAB_R[6]  ; FPA AT 06A.
U 10FD, 0000,007C,0180,0A30,0000,10FE  ;2086  LAB_R[6],CPSYNC  ; LOAD AR
U 10FE, 0000,003C,0180,0A38,0000,10FF  ;2087  LAB_R[7]  ; FPA AT 06C.
U 10FF, 0000,007C,0180,0A38,0000,1109  ;2088  LAB_R[7],CPSYNC  ; LOAD AR0
U 1109, 0000,003C,0180,0A40,0000,110A  ;2089  LAB_R[8]  ; FPA AT 06D.
U 110A, 0000,007C,0180,0A40,0000,110B  ;2090  LAB_R[8],CPSYNC  ; LOAD BR
U 110B, 0000,003C,0180,0A48,0000,1110  ;2091  LAB_R[9]  ; FPA AT 083.
U 1110, 0000,007C,0180,0A48,0000,1111  ;2092  LAB_R[9],CPSYNC  ; LOAD BR0
U 1111, 0000,003E,0180,0800,0000,0001  ;2093  RETURN1   ; FPA AT 028. LOAD COMPLETE.
            ;2094 ;*
            ;2095 ; THIS ROUTINE INITIALIZES THE MULTIPLIER SEQUENCER,AND THEN TRAPS
            ;2096 ; TO THE FPA UCODE THAT PERFORMS THE MULTIPLY.IT THEN JUMPS TO THE
            ;2097 ; ROUTINE THAT READS THE PRODUCT BACK TO THE CPU.
            ;2098 ;-
            ;2099
            ;2100 =0
U 1018, 0000,003D,0180,0800,0000,10E5  ;2101 INI.MUL: CALL[TRP.CE]
U 1019, 0000,003C,0180,0800,0000,1112  ;2102  NOP   ; FPA AT 0CE. INIT THE MULTIPLIER
U 1112, 0000,003C,0180,0800,0000,101A  ;2103  NOP   ; FPA AT 028.
U 101A, 0000,003D,0180,0800,0000,10E3  ;2104 =0 CALL[TRP.C8]
U 101B, 0000,003C,0180,0800,0000,1113  ;2105  NOP   ; FPA AT 0C8. START MULTIPLY
U 1113, 0000,003C,0180,0800,0000,1114  ;2106  NOP   ; FPA AT 0C9. CONTINUE MULTIPLY
U 1114, 0000,003C,0180,0800,0000,1115  ;2107  NOP   ; FPA AT 0CA. CONTINUE MULTIPLY
U 1115, 0000,003C,0180,0800,0000,1030  ;2108  J/PROD1   ; FPA AT 028.
            ;2109
            ;2110 ; *
            ;2111 ; THIS ROUTINE LOADS THE IB WITH A MUL DOUBLE OPCODE, IT THEN TAKES
            ;2112 ; THE MULTIPLIER AND MULTIPLICAND OPERANDS PREVIOUSLY LOADED IN THE FAD
            ;2113 ; AR AND BR LATCHES,AND LOADS THE MULTIPLIER OPERAND REGISTERS FOR A
            ;2114 ; MULTIPLY DOUBLE. THE FPA IS THEN TRAPPED TO THE UCODE THAT STARTS
            ;2115 ; THE MULTIPLY OPERATION AND STALLS AT UPC 028. THE ROUTINE THEN
```

```
            ;2116 ; JUMPS TO THE WAIT ROUTINE WHILE THE MULTIPLIER GENERATES THE PRODUCT.
            ;2117 ; THIS ROUTINE UTILIZES A DIFFERENT OPLD FIELD THAN THE MULD2
            ;2118 ; ROUTINE.
            ;2119 ;-
            ;2120
            ;2121 =0
U 101C. 2018.0039.6580.0800.4200.10F2  ;2122 MULD1: ALU_K[.10],FLUSH.IB,CALL[IBLOAD] ; MULD R,R
U 101D. 0000.003C.0180.0800.0000.1020  ;2123  NOP
U 1020. 0000.003D.0180.0800.0000.1012  ;2124 =0 CALL[TRP.E6]
U 1021. 0000.003C.0180.0800.0000.1116  ;2125  NOP    ; FPA AT 0E6.
            ;2126       ; MC1_AR1,MC0_AR0
U 1116. 0000.003C.0180.0800.0000.1117  ;2127  NOP    ; FPA AT 0E7.
            ;2128       ; MP1_BR1,MP0_BR0
U 1117. 0000.003C.0180.0800.0000.1118  ;2129  NOP    ; FPA AT 0C8. START MULTIPLY
U 1118. 0000.003C.0180.0800.0000.1119  ;2130  NOP    ; FPA AT 0C9. CONTINUE MULTIPLY
U 1119. 0000.003C.0180.0800.0000.111A  ;2131  NOP    ; FPA AT 028. CONTINUE MULTIPLY
U 111A. 0000.003C.0180.0800.0000.102C  ;2132  J/WAIT1    ; GO WAIT FOR MULTIPLY DONE?
            ;2133
            ;2134 ; *
            ;2135 ; THIS ROUTINE LOADS THE IB WITH A MUL DOUBLE OPCODE, IT THEN TAKES
            ;2136 ; THE MULTIPLIER AND MULTIPLICAND OPERANDS PREVIOUSLY LOADED IN THE FAD
            ;2137 ; AR AND BR LATCHES,AND LOADS THE MULTIPLIER OPERAND REGISTERS FOR A
            ;2138 ; MULTIPLY DOUBLE. THE FPA IS THEN TRAPPED TO THE UCODE THAT STARTS
            ;2139 ; THE MULTIPLY OPERATION AND STALLS AT UPC 028. THE ROUTINE THEN
            ;2140 ; JUMPS TO THE WAIT ROUTINE WHILE THE MULTIPLIER GENERATES THE PRODUCT.
            ;2141 ; THIS ROUTINE UTILIZES A DIFFERENT OPLD FIELD THAN THE MULD1
            ;2142 ; ROUTINE.
            ;2143 ;-
            ;2144
            ;2145 =0
U 1022. 2018.0039.6580.0800.4200.10F2  ;2146 MULD2: ALU_K[.10],FLUSH.IB,CALL[IBLOAD] ; MULD R,R
U 1023. 0000.003C.0180.0800.0000.1024  ;2147  NOP
U 1024. 0000.003D.0180.0800.0000.10E1  ;2148 =0 CALL[TRP.C3]
U 1025. 0000.003C.0180.0800.0000.111B  ;2149  NOP    ; FPA AT 0C3. MP0_AR0
U 111B. 0000.003C.0180.0800.0000.111C  ;2150  NOP    ; FPA AT 0C4. MP1_AR1
U 111C. 0000.003C.0180.0800.0000.111D  ;2151  NOP    ; FPA AT 0C5. MC0_BR0
U 111D. 0000.003C.0180.0800.0000.111E  ;2152  NOP    ; FPA AT 0C6. MC1_BR1
U 111E. 0000.003C.0180.0800.0000.111F  ;2153  NOP    ; FPA AT 0C8. START MULTIPLY
U 111F. 0000.003C.0180.0800.0000.1120  ;2154  NOP    ; FPA AT 0C9. CONTINUE MULTIPLY
U 1120. 0000.003C.0180.0800.0000.1121  ;2155  NOP    ; FPA AT 028. CONTINUE MULTIPLY
U 1121. 0000.003C.0180.0800.0000.102C  ;2156  J/WAIT1    ; GO TO WAIT FOR MULTIPLY DONE?
            ;2157
            ;2158 ;*
            ;2159 ; THE IB IS LOADED WITH AN EMODD OPCODE. THE FPA IS THEN TRAPPED TO
            ;2160 ; THE UCODE THAT LOADS THE OPERAND REGISTERS. A JUMP IS THEN TAKEN
            ;2161 ; TO THE ROUTINE THAT INITIATES THE MULTIPLY OPERATION.
            ;2162 ;-
            ;2163
            ;2164 =0
U 1028. 2018.0039.7D80.0800.4200.10F2  ;2165 EMODD: ALU_K[.18],FLUSH.IB,CALL[IBLOAD]
U 1029. 0000.003C.0180.0800.0000.102A  ;2166  NOP
U 102A. 0000.003D.0180.0800.0000.1012  ;2167 =0 CALL[TRP.E6]
U 102B. 0000.003C.0180.0800.0000.1122  ;2168  NOP    ; FPA AT 0E6.
            ;2169       ; MC1_AR1,MC0_AR0
U 1122. 0000.003C.0180.0800.0000.1123  ;2170  NOP    ; FPA AT 0E7.
```

```
U 1123, 0000,003C,0180,0800,0000,1018  ;2171  J/INI.MUL   ; MP1_BR1,MP0_BR0
            ;2172
            ;2173 ;*
            ;2174 ; THE WAIT1 ROUTINE TRAPS THE FPA TO UCODE WHICH WAITS FOR THE MULTIPLY
            ;2175 ; DONE SIGNAL FROM THE MULTIPLIER. UPOM THE RECEIPT OF THE MULTIPLY DONE,
            ;2176 ; THE ROUTINE READS BACK THE NORMALIZED AND UNNORMALIZED PRODUCT TO
            ;2177 ; THE CPU FOR VERIFICATION.
            ;2178 ;-
            ;2179
            ;2180 =0
U 102C, 0000,003D,0180,0800,0000,1010  ;2181 WAIT1: CALL[TRP.D0]
U 102D, 0000,063C,0180,0800,0000,1006  ;2182 MD1: ACC.SYNC?  ; WAIT FOR MULTIPLY DONE
U 1006, 0000,003C,0180,0800,0000,102D  ;2183 =110 J/MD1    ;
U 1007, 0000,003C,01D8,0800,0000,1124  ;2184  Q_ACC    ; FPA AT 0D1.
            ;2185      ; SAVE PRODUCT HI AND LO
            ;2186      ; IN AR LATCH AND IN
            ;2187      ; NORMALIZER REGISTER
U 1124, 0001,207C,0180,0988,0000,1125  ;2188  RC[1]_Q,CPSYNC  ; SAVE UNNORMALIZED PRODUCT HI
U 1125, 0000,003C,01D8,0800,0000,1126  ;2189  Q_ACC    ; FPA AT 0D2
U 1126, 0001,207C,0180,0980,0000,1127  ;2190  RC[0]_Q,CPSYNC  ; SAVE UNNORMALIZED PRODUCT LO
U 1127, 0000,003C,01D8,0800,0000,1128  ;2191  Q_ACC    ; FPA AT 0D3.
            ;2192      ; READ NORMALIZED PRODUCT LO
U 1128, 0A00,003C,0180,0800,0000,1129  ;2193  D_ACC    ; FPA AT 0D4,
U 1129, 0000,003E,0180,0800,0000,0001  ;2194  RETURN1    ; READ NORMALIZED PROCUCT HI
            ;2195
            ;2196 ;*
            ;2197 ; PROD1 TRAPS TO THE UCODE THAT CONTROLS THE SYNCHRONIZED
            ;2198 ; TRANSFER OF INTEGER PRODUCT HI AND LO TO THE CPU FOR
            ;2199 ; VERIFICATION.
            ;2200 ;-
            ;2201
            ;2202 =0
U 1030, 0000,003D,0180,0800,0000,10E8  ;2203 PROD1: CALL[TRP.CC]
U 1031, 0000,007C,01D8,0800,0000,112A  ;2204  Q_ACC,CPSYNC  ; FPA AT 0CC.
            ;2205      ; BUSA_INTEGER PRODUCT HI
U 112A, 0A00,007C,0180,0800,0000,112B  ;2206  D_ACC,CPSYNC  ; FPA AT 0CD.
            ;2207      ; BUSA_INTEGER PRODUCT LO
U 112B, 0000,003C,0180,0800,0000,112C  ;2208  NOP    ; FPA AT 0CE.
            ;2209      ; INITIALIZE MULTIPLIER LOGIC
U 112C, 0000,003E,0180,0800,0000,0001  ;2210  RETURN1    ; FPA AT 028.
            ;2211
            ;2212 =0
U 1032, 0000,003D,0180,0800,0000,10EF  ;2213 FAD.CHK: CALL[TRP.FB]
U 1033, 0000,003C,01D8,0800,0000,112D  ;2214  Q_ACC    ; FPA AT 0FB. BUSA_AR1
U 112D, 0001,207C,0180,09F0,0000,112E  ;2215  RC[0E]_Q,CPSYNC  ; SAVE
U 112E, 0A00,003C,0180,0800,0000,112F  ;2216  D_ACC    ; FPA AT 0FC. BUSA_AR0
U 112F, 0001,007C,0180,09E8,0000,1130  ;2217  RC[0D]_D,CPSYNC  ; SAVE
U 1130, 0000,003C,01D8,0800,0000,1131  ;2218  Q_ACC    ; FPA AT 0FD. BUSA_BR1
U 1131, 0001,207C,0180,09E0,0000,1132  ;2219  RC[0C]_Q,CPSYNC  ; SAVE
U 1132, 0A00,003C,0180,0800,0000,1133  ;2220  D_ACC    ; FPA AT 0FE. BUSA_BR0
U 1133, 0001,003C,0180,09D8,0000,1134  ;2221  RC[0B]_D  ; SAVE
U 1134, 0000,003E,0180,0800,0000,0001  ;2222  RETURN1
            ;2223
            ;2224 ; THIS ROUTINE IS CALLED TO TAKE AN ADDRSSS IN THE D REGISTER AND
            ;2225 ; CONVERT IT TO A FORM WHICH WHEN LOADED INTO ID[ACC.2] WILL SET UP
```

```
            ;2226 ; THE FPA FOR A TRAP TO THAT ADDRESS.
            ;2227
U 1135. 0000.003C.65F8.0800.0084.7136  ;2228 GENTRA: SC_K[.10],Q_0
U 1136. 0D00.003C.8D80.0800.0084.7137  ;2229  D_DAL.SC,SC_K[.1F]
            ;2230
            ;2231
            ;2232 ;
            ;2233 ; THIS SUBROUTINE IS CALLED TO PACK A FRACTION (HI BITS <63:32>
            ;2234 ; ASSUMED TO BE IN R[4] AND LO BIT2 <31:00> IN R[5])
            ;2235 ; INTO THE FLOATING FORMAT AS IT WOULD APPEAR ON BUS A OR BUS B.
            ;2236 ; THE HIGH RESULT IS LEFT IN R[6] AND THE LO RESULT IN R[7]
            ;2237 ;
U 1137. 0000.003C.01C0.0A28.0000.1138  ;2238 UNSEQA: Q_R[5]   ; GET FRACTION INTO D AND Q.
U 1138. 0800.003C.B980.0A20.0084.7139  ;2239  D_R[4],SC_K[.19]
U 1139. 0D00.003C.0180.0800.0000.113A  ;2240  D_DAL.SC
U 113A. 0000.003C.65E0.0800.0084.713B  ;2241  Q_D,SC_K[.10]
U 113B. 0D00.003C.0180.0800.0000.113C  ;2242  D_DAL.SC
U 113C. 0001.003C.0180.0AB8.0000.113D  ;2243  R[7]_D   ; SAVE LEAST SIGNIFICANT RESULT, LO
U 113D. 0800.003C.01C0.0A20.0000.113E  ;2244  D_R[4],Q_R[4]
U 113E. 0819.0024.5980.0800.0000.113F  ;2245  ALU_D[ANDNOT]K[.7F],D_ALU
U 113F. 0100.003C.5D88.0800.0084.7140  ;2246  D_D.LEFT2,Q_Q.LEFT2,SI/ZERO,SC_K[.7]
U 1140. 0D00.003C.0180.0800.0000.1141  ;2247  D_DAL.SC
U 1141. 0001.003C.0180.0AB0.0084.7142  ;2248  R[6]_D,SC_K[.8]
U 1142. 0818.0034.59F8.0A28.0000.1143  ;2249  LAB_R[5],ALU_LA[AND]K[.7F],D_ALU,Q_0
U 1143. 0D00.003C.01C0.0A30.0000.1144  ;2250  D_DAL.SC,Q_R[6]
U 1144. 001D.0032.0180.0AB0.0000.0001  ;2251  ALU_D[OR]Q,R[6]_ALU,RETURN1 ; SAVE HI RESULT.
            ;2252 ;
            ;2253 ; THIS SUBROUTINE IS CALLED TO INITIALIZE THE FPA. THIS IS DONE BY:
            ;2254 ; 1 TRAPPING THE FPA TO LOCATION 28, WHICH CONTAINS
            ;2255 ;   28: NAD/28   ; NOP
            ;2256 ; 2 LOADING A HALT INSTRUCTION INTO THE IB
            ;2257 ; 3 TRAPPING THE FPA TO LOCATION 0 THERE BY PUTTING
            ;2258 ;  THE FPA INTO ITS IRD STATE AND CAUSING THE FPA TO
            ;2259 ;  INITIALIZE ITSELF.
            ;2260 ; 4 TRAPPING THE FPA BACK TO LOCATION 28.
            ;2261 ;
U 1145. 0818.0038.4580.0800.0000.1146  ;2262 FPINIT: D_K[.8000]
U 1146. 0000.003C.5D80.3C00.0000.1034  ;2263  ID[ACC.CS]_D   ; TURN ON FPA
U 1034. 0818.0039.2D80.0800.0000.1135  ;2264 =0 D_K[.28],CALL,J/GENTRA ; CALCULATE THE ADDRESS OF THE FPA UCODE.
U 1035. 0000.003C.5980.3C00.0000.1147  ;2265  ID[ACC.2]_D
U 1147. 0000.00FC.0380.0800.0000.1036  ;2266  TRAP.FPA   ; TRAP FPA TO 28.
U 1036. 0018.0039.1980.0800.0200.114D  ;2267 =0 VA_K[ZERO],CALL,J/LOADIB ; LOAD HALT INTO IB.
U 1037. 0F00.003C.0180.0800.0000.1038  ;2268  D_0
U 1038. 0000.003D.0180.0800.0000.1135  ;2269 =0 CALL,J/GENTRA
U 1039. 0000.003C.5980.3C00.0000.1148  ;2270  ID[ACC.2]_D
U 1148. 0000.00FC.0380.0800.0000.103A  ;2271  TRAP.FPA   ; TRAP FPA TO 0
U 103A. 0818.0039.2D80.0800.0000.1135  ;2272 =0 D_K[.28],CALL,J/GENTRA
U 103B. 0000.003C.5980.3C00.0000.1149  ;2273  ID[ACC.2]_D
U 1149. 0000.00FE.0380.0800.0000.0001  ;2274  TRAP.FPA,RETURN1 ; TRAP FPA TO 28.
            ;2275 ;
            ;2276 ;
            ;2277 ; HALT INSTRUCTION IN CACHE WHICH IS LOADED INTO IB:
            ;2278 ;%0
            ;2279 ;$ 0000.0000. 0000.0000
            ;2280 ;
```

```
            ;2281 ;
            ;2282 ; THIS ROUTINE IS CALLED TO PUT THE FPA INTO IRD. THIS IS ACCOMPLISHED
            ;2283 ; BY TRAPPING THE FPA TO LOCATION 0.
            ;2284 ;
            ;2285 =0
U 103C, 0F00.003D.0180.0800.0000.1135  ;2286 FPIRD: D_0,CALL,J/GENTRA
U 103D, 0000.003C.5980.3C00.0000.114A  ;2287  ID[ACC.2]_D
U 114A, 0000.00FC.0380.0800.0000.114B  ;2288  TRAP.FPA  ; TRAP TO FPA 0
U 114B, 0000.003C.0180.0800.0000.114C  ;2289  NOP
U 114C, 0000.003E.0180.0800.0000.0001  ;2290  RETURN1
            ;2291 ;
            ;2292 ; THIS ROUTINE IS CALLED TO LOAD DATA INTO THE INSTRUCTION BUFFER.
            ;2293 ; BEFORE BEING CALLED THE DATA SHOULD BE A HIT IN THE CACHE AND THE
            ;2294 ; ADDRESS OF THE DATA SHOULD BE IN THE VA.
            ;2295 ;
U 114D, 2000.003C.0180.0800.0000.114E  ;2296 LOADIB: IBC/FLUSH  ; CLEAR THE IB
U 114E, 3000.003C.0180.6000.0000.114F  ;2297  MCT/READ.V.NEWPC,IBC/START ; LOAD IPA AND START.
U 114F, 0000.003C.0180.F800.0000.1150  ;2298 IBW: MCT/ALLOW.IB.READ ; WAIT FOR IB TO FILL.
U 1150, 0000.003C.0180.F800.0000.1151  ;2299  MCT/ALLOW.IB.READ
U 1151, 0000.003C.0180.F800.0000.1152  ;2300  MCT/ALLOW.IB.READ
U 1152, 0000.003C.0180.F800.0000.1153  ;2301  MCT/ALLOW.IB.READ
U 1153, 0000.003C.0180.F800.0000.1154  ;2302  MCT/ALLOW.IB.READ
U 1154, 1000.003C.0180.F800.0000.1156  ;2303  MCT/ALLOW.IB.READ,IBC/STOP ; SHOULD BE FULL NOW.
U 1156, 0000.003E.0180.0800.0000.0001  ;2304  RETURN1
U 1157, 3000.003C.0180.0800.0000.114F  ;2305 WAITIB: IBC/START,J/IBW
            ;2306
            ;2307 ;*
            ;2308 ; THIS ROUTINE GENERATES A 4 BIT PATTERN STARTING AT 0000 AND
            ;2309 ; INCREMENTING TO FFFF. THE PATTERN IS SHIFTED INTO EACH NIBBLE
            ;2310 ; OF THE D REGISTER. THE Q AND D REGISTERS RETURN THE PATTERN TO
            ;2311 ; THE CALLING PROGRAM. R2 IS LOADED FROM THE CALLING PROGRAM
            ;2312 ; WITH A NUMBER FROM 0 TO F, DEPENDING ON WHICH MULTIPLICAND
            ;2313 ; IS NEEDED.
            ;2314 ;-
            ;2315
            ;2316
U 1158, 0000.003C.01C0.0A10.0000.1159  ;2317 MCAND: Q_R[2]
U 1159, 0000.003C.ED80.0800.0084.715A  ;2318  SC_K[.1B]
U 115A, 0C00.003C.05F8.0800.0084.915B  ;2319  SC_SC+K[.1],D_Q,Q_0
U 115B, 0D00.003C.0180.0800.0000.115C  ;2320  D_DAL.SC
U 115C, 0F00.003C.11E0.0800.0084.715D  ;2321  Q_D,SC_K[.4],D_0
U 115D, 0D00.003C.0180.0800.0000.115E  ;2322  D_DAL.SC
U 115E, 0D00.003C.0180.0800.0000.115F  ;2323  D_DAL.SC
U 115F, 0D00.003C.0180.0800.0000.1160  ;2324  D_DAL.SC
U 1160, 0D00.003C.0180.0800.0000.1161  ;2325  D_DAL.SC
U 1161, 0D00.003C.0180.0800.0000.1162  ;2326  D_DAL.SC
U 1162, 0D00.003C.0180.0800.0000.1163  ;2327  D_DAL.SC
U 1163, 0D00.003C.0180.0800.0000.1164  ;2328  D_DAL.SC
U 1164, 0D00.003C.0180.0800.0000.1165  ;2329  D_DAL.SC
U 1165, 0000.003E.01E0.0800.0000.0001  ;2330  Q_D,RETURN1
            ;2331
            ;2332
```

```
;2333  .PAGE "TEST 2A1 EMODD FLOATING ONE BY 0.1"
;2334 ;*****************************************************************
;2335 ;**
;2336 ;
;2337 ; EMODD, FLOATING ONE BY 0.1
;2338 ;
;2339 ; TEST DESCRIPTION
;2340 ;
;2341 ; THIS TEST DOES A 56X64 BIT MULTIPLY WITH A FLOATING ONE PATTERN
;2342 ; IN THE MULTIPLICAND (MC) AND MULTIPLICAND EXTENSION (MCX), AND
;2343 ; WITH MULTIPLIER) CONTAINING ONLY THE HIDDEN BIT SET.
;2344 ;
;2345 ; TEST SEQUENCE   DESCRIPTION
;2346 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2347 ; 2  PACK THE DATA
;2348 ; 3  PERFORM THE MULTIPLY
;2349 ; 4  VERIFY RESULTS
;2350 ;
;2351 ; FPA UCODE USED
;2352 ;
;2353 ; LOCATION CONTENTS
;2354 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2355 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2356 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2357 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2358 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2359 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2360 ; 028: NAD/028
;2361 ; 0E6: FADC/A,BSC/FAL.HL,OPLD/MC,NAD/0E7
;2362 ; 0E7: FSDC/B,BSC/FAL.LH,OPLD/MP,NAD/0C8
;2363 ; 0CE: OPLD/INIT,NAD/028
;2364 ; 028: NAD/028
;2365 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2366 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2367 ; 0CA: NAD/028
;2368 ; 028: NAD/028
;2369 ; 0CC: BSC/INTH,CPSYNC,WAIT,NAD/0CD
;2370 ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;2371 ; 0CE: OPLD/INIT,NAD/028
;2372 ;
;2373 ; LOGIC DESCRIPTION
;2374 ;
;2375 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTILPIER
;2376 ; LOGIC ARE ON THE FMH AND FML LOGIC MODULES.
;2377 ;
;2378 ; ERROR DESCRIPTION
;2379 ;
;2380 ; EXPECTED INTEGER PRODUCT HI
;2381 ; RECEIVED INTEGER PRODUCT HI
;2382 ; EXPECTED INTEGER PRODUCT LO
;2383 ; RECEIVED INTEGER PRODUCT LO
;2384 ; UNPACKED DATA HI
;2385 ; UNPACKED DATA LO
;2386 ; PACKED MULTIPLIER HI (MP1)
;2387 ; PACKED MULTIPLIER LO (MP0)
```

```
            ;2388 ; PACKED MULTIPLICAND HI (MC1)
            ;2389 ; PACKED MULTIPLICAND LO (MC0)
            ;2390 ; BIT COUNTER
            ;2391 ;
            ;2392 ;
            ;2393 ;
            ;2394 ;
            ;2395 ; SYNC POINT
            ;2396 ;
            ;2397 ;
            ;2398 ;
            ;2399 ;--
            ;2400 ;
            ;2401 ;*****************************************************************
            ;2402
            ;2403
            ;2404
            ;2405 ;/////////////////////////////////////////////////
            ;2406 ;+
            ;2407 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
            ;2408 ;-
            ;2409
            ;2410
            ;2411 =0
U 1040, 0018,0039,F580,09F8,0000,10BC  ;2412 MULT7: NEWTST[.A]
U 1041, 0810,0038,0180,0978,0000,1166  ;2413  D_RC[0F]
U 1166, 0019,0014,0580,09F8,0000,1167  ;2414  ALU_D+K[.1],RC[0F]_ALU
U 1167, 0003,003C,0180,0A80,0000,1168  ;2415  R[0]_0    ; PATTERN COUNTER
U 1168, 0818,0038,4D80,0800,0000,1169  ;2416  D_K[.FF00]
U 1169, 0819,0030,4180,0800,0000,116A  ;2417  D_D.OR.K[.80]
U 116A, 0001,0028,0180,0A88,0000,116B  ;2418  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 116B, 0818,0038,7980,0800,0000,116C  ;2419  D_K[.30]
U 116C, 0819,0030,8980,0800,0000,116D  ;2420  D_D.OR.K[.D]
U 116D, 0001,003C,0180,09A0,0000,1042  ;2421  RC[4]_D    ; BIT COUNT OF 61
U 1042, 0000,003D,0180,0800,0000,10CE  ;2422 =0 ERLOOP
U 1043, 0810,0038,0180,0920,0082,116E  ;2423 MUL11: D_RC[4],SC_RC[4] ; D GETS  COUNT, SC GETS PATTERN
U 116E, 0019,0034,7580,0800,0010,116F  ;2424  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING MULTIPLICAND
            ;2425       ; LO WITH PATTERN, MULTIPLICAND HI
            ;2426       ; WITH ZERO.
U 116F, 0000,013C,0180,0800,0000,1044  ;2427  Z?
U 1044, 0000,003C,0180,0800,0000,1171  ;2428 =0 J/MUL12
U 1045, 0003,001C,0180,0AA8,0000,1170  ;2429  R[5]_NOT.MASK  ; FLOATING 1
U 1170, 0003,003C,0180,0AA0,0000,1173  ;2430  R[4]_0,J/MUL13
U 1171, 0003,003C,0180,0AA8,0000,1172  ;2431 MUL12: R[5]_0    ; MULTIPLICAND LO GETS 0
U 1172, 0003,001C,0180,0AA0,0000,1173  ;2432  R[4]_NOT.MASK  ; FLOATING 1
            ;2433
            ;2434 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;2435
U 1173, 0800,003C,0180,0A20,0000,1174  ;2436 MUL13: D_R[4]
U 1174, 0001,003C,0180,09D0,0000,1175  ;2437  RC[0A]_D  ; UNPACKED DATA HI
U 1175, 0000,003C,5180,0800,0084,7176  ;2438  SC_K[.1E]
U 1176, 0003,001C,01C0,0800,0000,1177  ;2439  Q_NOT.MASK
U 1177, 001D,0030,0180,09F0,0000,1178  ;2440  RC[0E]_D.OR.Q  ; EXPECTED INTEGER PRODUCT HI
U 1178, 0800,003C,0180,0A28,0000,1179  ;2441  D_R[5]
U 1179, 0001,003C,0180,09E0,0000,117A  ;2442  RC[0C]_D  ; EXPECTED INTEGER DATA LO
```

```
U 117A, 0001,003C,0180,09C8,0000,1046  ;2443  RC[9]_D   ; UNPACKED DATA LO
          ;2444
          ;2445 ; PACK OPERANDS IN R4, R5 FOR MULTIPLY OPERATION
          ;2446
U 1046, 0000,003D,0180,0800,0000,1137  ;2447 =0 CALL[UNSEQA]
U 1047, 0800,003C,0180,0A30,0000,117B  ;2448  D_R[6]   ;
U 117B, 080D,0034,0180,0A08,0000,117C  ;2449  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 117C, 0001,003C,0180,0AB0,0000,117D  ;2450  R[6]_D   ; MULTIPLICAND HI MASKED
U 117D, 0001,003C,0180,09B0,0000,117E  ;2451  RC[6]_D   ; SAVE
U 117E, 0003,003C,0180,0AC0,0000,117F  ;2452  R[8]_0   ; MULTIPLIER HI
U 117F, 0003,003C,0180,09C0,0000,1180  ;2453  RC[8]_0   ; SAVE
U 1180, 0810,0038,D580,0920,0000,1181  ;2454  D_RC[4],KMX/.6
U 1181, 0019,0000,D580,0800,0010,1182  ;2455  ALU_D-K[.6],CLK.UBCC
U 1182, 0000,013C,0180,0800,0000,1048  ;2456  Z?
U 1048, 0000,003C,0180,0800,0000,1183  ;2457 =0 J/NEG
U 1049, 0000,003C,0180,0800,0000,101F  ;2458  J/MCI   ; BIT COUNT =6
U 1183, 0000,1B3C,0180,0800,0000,1017  ;2459 NEG: ALU.N?   ; BIT COUNT <6?
U 1017, 0000,003C,0180,0800,0000,1186  ;2460 =0111 J/MC0   ; BIT COUNT NOT <6
U 101F, 0800,003C,0180,0A28,0000,1184  ;2461 MCI: D_R[5]
U 1184, 0821,003C,0180,0800,0000,1185  ;2462  ALU_D,D_ALU.LEFT ; UPDATE EXPECTED INTEGER PRODUCT LO
U 1185, 0001,003C,0180,0AB8,0000,1186  ;2463  R[7]_D   ; MCI LO 8 BITS
U 1186, 0800,003C,0180,0A38,0000,1187  ;2464 MC0: D_R[7]   ; MULTIPLICAND LO
U 1187, 0001,003C,0180,09A8,0000,1188  ;2465  RC[5]_D   ; SAVE
U 1188, 0003,003C,0180,0AC8,0000,1189  ;2466  R[9]_0   ; MULTIPLIER LO
U 1189, 0003,003C,0180,09B8,0000,104A  ;2467  RC[7]_0   ; SAVE
          ;2468
          ;2469 ; LOAD OPERANDS, MULTIPLY
          ;2470
U 104A, 0000,003D,0180,0800,0000,1014  ;2471 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
U 104B, 0000,003C,0180,0800,0000,104C  ;2472  NOP
U 104C, 0000,003D,0180,0800,0000,1028  ;2473 =0 CALL[EMODD]  ; EMODD
U 104D, 0001,203C,0180,09E8,0000,118A  ;2474  RC[0D]_Q   ; SAVE INTEGER PRODUCT HI
U 118A, 0001,003C,0180,09D8,0000,118B  ;2475  RC[0B]_D   ; SAVE INTEGER PRODUCT LO
U 118B, 0810,0038,5D80,0920,0000,118C  ;2476  D_RC[4],KMX/.7
U 118C, 0019,0000,5D80,0800,0010,118D  ;2477  ALU_D-K[.7],CLK.UBCC
U 118D, 0000,013C,0180,0800,0000,1050  ;2478  Z?
U 1050, 0000,003C,0180,0800,0000,118E  ;2479 =0 J/NEG1
U 1051, 0000,003C,0180,0800,0000,102F  ;2480  J/MC11   ; BIT COUNT =7
U 118E, 0000,1B3C,0180,0800,0000,1027  ;2481 NEG1: ALU.N?   ; BIT COUNT <7?
U 1027, 0000,003C,0180,0800,0000,1190  ;2482 =0111 J/MC01   ; BIT COUNT NOT =7
U 102F, 0810,0038,0180,0958,0000,118F  ;2483 MC11: D_RC[0B]
U 118F, 0019,0034,4980,09D8,0000,1192  ;2484  ALU_D.AND.K[.FF],RC[0B]_ALU,J/NXT ; PASS MCX BITS ONLY
U 1190, 0810,0038,0180,0958,0000,1191  ;2485 MC01: D_RC[0B]
U 1191, 0019,0024,4980,09D8,0000,1192  ;2486  ALU_D[ANDNOT]K[.FF],RC[0B]_ALU ; PASS MC0 BITS ONLY
U 1192, 0011,2020,0180,0970,0010,1193  ;2487 NXT: ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 1193, 0000,013C,0180,0800,0000,1052  ;2488  Z?
U 1052, 0000,003D,0180,0800,0000,10CF  ;2489 =0 ERROR1
U 1053, 0010,0038,01C0,0958,0000,1194  ;2490  Q_RC[0B]
U 1194, 0011,2020,0180,0960,0010,1195  ;2491  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1195, 0000,013C,0180,0800,0000,1054  ;2492  Z?
U 1054, 0000,003D,0180,0800,0000,10CF  ;2493 =0 ERROR1
U 1055, 0810,0038,0180,0920,0010,1196  ;2494  ALU_RC[4],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 1196, 0000,013C,0180,0800,0000,1056  ;2495  Z?
U 1056, 0019,0000,0580,09A0,0000,1043  ;2496 =0 ALU_D-K[.1],RC[4]_ALU,J/MUL11 ; DECREMENT BIT COUNTER
U 1057, 0000,003C,0180,0800,0000,1058  ;2497  NOP
```

```
:2498
:2499
```

```
;2500  .PAGE "TEST 2A2 EMODD 0.1 BY FLOATING ONE"
;2501 ;*****************************************************************
;2502 ;**
;2503 ;
;2504 ; EMODD, 0.1 BY FLOATING ONE
;2505 ;
;2506 ; TEST DESCRIPTION
;2507 ;
;2508 ; THIS TEST DOES A 56X64 BITULTIPLY WITH ONLY THE HIDDEN BIT SET
;2509 ; IN THE MULTIPLICAND (MC) AND MULTIPLICAND EXTENSION (MCX), AND
;2510 ; WITH MULTIPLIER CONTAINING A FLOATING ONE PATTERN.
;2511 ;
;2512 ;
;2513 ; TEST SEQUENCE   DESCRIPTION
;2514 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2515 ; 2  PACK THE DATA
;2516 ; 3  PERFORM THE MULTIPLY
;2517 ; 4  VERIFY RESULTS
;2518 ;
;2519 ; FPA UCODE USED
;2520 ;
;2521 ; LOCATION CONTENTS
;2522 ;
;2523 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2524 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2525 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2526 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2527 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2528 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2529 ; 028: NAD/028
;2530 ; 0E6: FADC/A,BSC/FAL.HL,OPLD/MC,NAD/0E7
;2531 ; 0E7: FSDC/B,BSC/FAL.LH,OPLD/MP,NAD/0C8
;2532 ; 0CE: OPLD/INIT,NAD/028
;2533 ; 028: NAD/028
;2534 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2535 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2536 ; 0CA: NAD/028
;2537 ; 028: NAD/028
;2538 ; 0CC: BSC/INTH,CPSYNC,WAIT,NAD/0CD
;2539 ; 0CD: BSC/INTL,FPSYNC,WAIT,NAD/0CE
;2540 ; 0CE: OPLD/INIT,NAD/028
;2541 ;
;2542 ; LOGIC DESCRIPTION
;2543 ;
;2544 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTILPIER
;2545 ; LOGIC ARE ON THE FMH AND FML LOGIC MODULES.
;2546 ;
;2547 ;
;2548 ;
;2549 ;
;2550 ;
;2551 ; ERROR DESCRIPTION
;2552 ;
;2553 ; EXPECTED INTEGER PRODUCT HI
;2554 ; RECEIVED INTEGER PRODUCT HI
```

```
            ;2555 ; EXPECTED INTEGER PRODUCT LO
            ;2556 ; RECEIVED INTEGER PRODUCT LO
            ;2557 ; UNPACKED DATA HI
            ;2558 ; UNPACKED DATA LO
            ;2559 ; PACKED MULTIPLIER HI (MP1)
            ;2560 ; PACKED MULTIPLIER LO (MP0)
            ;2561 ; PACKED MULTIPLICAND HI (MC1)
            ;2562 ; PACKED MULTIPLICAND LO (MC0)
            ;2563 ; BIT COUNTER
            ;2564 ;
            ;2565 ;
            ;2566 ;
            ;2567 ; SYNC POINT
            ;2568 ;
            ;2569 ;
            ;2570 ;
            ;2571 ;--
            ;2572 ;
            ;2573 ;*****************************************************************
            ;2574
            ;2575
            ;2576
            ;2577 ;/////////////////////////////////////////////////
            ;2578 ;*
            ;2579 ; FLOAT A 1 THROUGH THE MULTIPLIER WITH MULTIPLICAND=0.1.
            ;2580 ;-
            ;2581
            ;2582
            ;2583 =0
U 1058. 0018.0039.F580.09F8.0000.10BC  ;2584 MULT8: NEWTST[.A]
U 1059. 0810.0038.0180.0978.0000.1197  ;2585  D_RC[0F]
U 1197. 0019.0014.0580.09F8.0000.1198  ;2586  ALU_D+K[.1],RC[0F]_ALU
U 1198. 0003.003C.0180.0A80.0000.1199  ;2587  R[0]_0    ; PATTERN COUNTER
U 1199. 0818.0038.4D80.0800.0000.119A  ;2588  D_K[.FF00]
U 119A. 0819.0030.4180.0800.0000.119B  ;2589  D_D.OR.K[.80]
U 119B. 0001.0028.0180.0A88.0000.119C  ;2590  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 119C. 0818.0038.7980.0800.0000.119D  ;2591  D_K[.30]
U 119D. 0819.0030.8980.0800.0000.119E  ;2592  D_D.OR.K[.D]
U 119E. 0001.003C.0180.09A0.0000.105C  ;2593  RC[4]_D    ; BIT COUNT OF 61
U 105C. 0000.003D.0180.0800.0000.10CE  ;2594 =0 ERLOOP
U 105D. 0810.0038.0180.0920.0082.119F  ;2595 MUL14: D_RC[4],SC_RC[4] ; D GETS  COUNT, SC GETS PATTERN
U 119F. 0019.0034.7580.0800.0010.11A0  ;2596  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING MULTIPLICAND
            ;2597       ; LO WITH PATTERN, MULTIPLICAND HI
            ;2598       ; WITH ZERO.
U 11A0. 0000.013C.0180.0800.0000.1060  ;2599  Z?
U 1060. 0000.003C.0180.0800.0000.11A3  ;2600 =0 J/MUL15
U 1061. 0803.001C.0180.0800.0000.11A1  ;2601  D_NOT.MASK  ; FLOATING 1
U 11A1. 0019.0024.4980.0AA8.0000.11A2  ;2602  ALU_D[ANDNOT]K[.FF],R[5]_ALU ; MASK OUT BITS <7:0>
U 11A2. 0003.003C.0180.0AA0.0000.11A5  ;2603  R[4]_0,J/MUL16
U 11A3. 0003.003C.0180.0AA8.0000.11A4  ;2604 MUL15: R[5]_0    ; MULTIPLICAND LO GETS 0
U 11A4. 0003.001C.0180.0AA0.0000.11A5  ;2605  R[4]_NOT.MASK  ; FLOATING 1
            ;2606
            ;2607 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;2608
U 11A5. 0800.003C.0180.0A20.0000.11A6  ;2609 MUL16: D_R[4]
```

```
U 11A6, 0001.003C.0180.09D0.0000.11A7  ;2610  RC[0A]_D  ; UNPACKED DATA HI
U 11A7, 0000.003C.5180.0800.0084.71A8  ;2611  SC_K[.1E]
U 11A8, 0003.001C.01C0.0800.0000.11A9  ;2612  Q_NOT.MASK
U 11A9, 001D.0030.0180.09F0.0000.11AA  ;2613  RC[0E]_D.OR.Q  ; EXPECTED INTEGER PRODUCT HI
U 11AA, 0800.003C.0180.0A28.0000.11AB  ;2614  D_R[5]
U 11AB, 0001.003C.0180.09E0.0000.11AC  ;2615  RC[0C]_D  ; UNPACKED DATA LO
U 11AC, 0001.003C.0180.09C8.0000.1062  ;2616  RC[9]_D   ; SAVE
          ;2617
          ;2618 ; PACK OPERANDS IN R4, R5 FOR MULTIPLY OPERATION
          ;2619
U 1062, 0000.003D.0180.0800.0000.1137  ;2620 =0 CALL[UNSEQA]
U 1063, 0800.003C.0180.0A30.0000.11AD  ;2621  D_R[6]   ;
U 11AD, 080D.0034.0180.0A08.0000.11AE  ;2622  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11AE, 0003.003C.0180.0AB0.0000.11AF  ;2623  R[6]_0   ; MULTIPLICAND HI MASKED
U 11AF, 0003.003C.0180.09B0.0000.11B0  ;2624  RC[6]_0   ; SAVE
U 11B0, 0001.003C.0180.0AC0.0000.11B1  ;2625  R[8]_D   ; MULTIPLIER HI
U 11B1, 0001.003C.0180.09C0.0000.11B2  ;2626  RC[8]_D   ; SAVE
U 11B2, 0800.003C.0180.0A38.0000.11B3  ;2627  D_R[7]   ;
U 11B3, 0003.003C.0180.0AB8.0000.11B4  ;2628  R[7]_0   ; MULTIPLICAND LO
U 11B4, 0001.003C.0180.09A8.0000.11B5  ;2629  RC[5]_D   ; SAVE
U 11B5, 0001.003C.0180.0AC8.0000.11B6  ;2630  R[9]_D   ; MULTIPLIER LO
U 11B6, 0001.003C.0180.09B8.0000.1064  ;2631  RC[7]_D   ; SAVE
          ;2632
          ;2633 ; LOAD OPERANDS, MULTIPLY
          ;2634
U 1064, 0000.003D.0180.0800.0000.1014  ;2635 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
U 1065, 0000.003C.0180.0800.0000.1066  ;2636  NOP
U 1066, 0000.003D.0180.0800.0000.1028  ;2637 =0 CALL[EMODD]  ; EMODD
U 1067, 0001.203C.0180.09E8.0000.11B7  ;2638  RC[0D]_Q   ; SAVE INTEGER PRODUCT HI
U 11B7, 0001.003C.0180.09D8.0000.11B8  ;2639  RC[0B]_D   ; SAVE INTEGER PRODUCT LO
U 11B8, 0011.2020.0180.0970.0010.11B9  ;2640  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 11B9, 0000.013C.0180.0800.0000.1068  ;2641  Z?
U 1068, 0000.003D.0180.0800.0000.10CF  ;2642 =0 ERROR1
U 1069, 0010.0038.01C0.0958.0000.11BA  ;2643  Q_RC[0B]
U 11BA, 0011.2020.0180.0960.0010.11BB  ;2644  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 11BB, 0000.013C.0180.0800.0000.106A  ;2645  Z?
U 106A, 0000.003D.0180.0800.0000.10CF  ;2646 =0 ERROR1
U 106B, 0810.0038.0180.0920.0010.11BC  ;2647  ALU_RC[4],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 11BC, 0000.013C.0180.0800.0000.106C  ;2648  Z?
U 106C, 0019.0000.0580.09A0.0000.105D  ;2649 =0 ALU_D-K[.1],RC[4]_ALU,J/MUL14 ; DECREMENT BIT COUNTER
U 106D, 0000.003C.0180.0800.0000.1070  ;2650  NOP
          ;2651
```

```
;2652  .PAGE "TEST 2A3 MULTIPLY DOUBLE FLOATING ONE BY 0.1"
;2653 ;*****************************************************************
;2654 ;**
;2655 ;
;2656 ; MULTIPLY DOUBLE, FLOATING ONE BY 0.1
;2657 ;
;2658 ; TEST DESCRIPTION
;2659 ;
;2660 ; THIS TEST SUCCESSIVELY MULTIPLIES A FLOATING (1 PATTERN
;2661 ; IN THE MULTIPLICAND REGISTER, WITH THE MULTIPLIER REGISTER
;2662 ; CONTAINING ONLY THE HIDDEN BIT SET.
;2663 ;
;2664 ;
;2665 ; TEST SEQUENCE   DESCRIPTION
;2666 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2667 ; 2  PACK THE DATA
;2668 ; 3  PERFORM THE MULTIPLY
;2669 ; 4  VERIFY RESULTS
;2670 ;
;2671 ; FPA UCODE USED
;2672 ;
;2673 ; LOCATION CONTENTS
;2674 ;
;2675 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2676 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2677 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2678 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2679 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2680 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2681 ; 028: NAD/028
;2682 ; 0E6: FADC/A,BSC/FAL.HL,OPLD/MC,NAD/0E7
;2683 ; 0E7: FSDC/B,BSC/FAL.LH,OPLD/MP,NAD/0C8
;2684 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2685 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2686 ; 0CA: NAD/028
;2687 ; 028: NAD/028
;2688 ; 0D0: BEN6,NAD/0D0,BMXC/LB,BSC,PQ
;2689 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2690 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2691 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2692 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2693 ;
;2694 ; LOGIC DESCRIPTION
;2695 ;
;2696 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2697 ; LOGIC ARE ON THE FMH AND FML LOGIC MODULES.
;2698 ;
;2699 ; ERROR DESCRIPTION
;2700 ;
;2701 ; PACKED EXPECTED NORMALIZED PRODUCT HI
;2702 ; PACKED RECEIVED NORMALIZED PRODUCT HI
;2703 ; PACKED EXPECTED NORMALIZED PRODUCT LO
;2704 ; PACKED RECEIVED NORMALIZED PRODUCT LO
;2705 ; PACKED EXPECTED UNNORMALIZED PRODUCT HI
;2706 ; PACKED RECEIVED UNNORMALIZED PRODUCT HI
```

```
            ;2707 ; PACKED EXPECTED UNNORMALIZED PRODUCT LO
            ;2708 ; PACKED RECEIVED UNNORMALIZED PRODUCT LO
            ;2709 ; UNPACKED DATA HI
            ;2710 ; UNPACKED DATA LO
            ;2711 ; PACKED MULTIPLIER REGISTER HI
            ;2712 ; PACKED MULTIPLIER REGISTER LO
            ;2713 ; BIT COUNTER
            ;2714 ;
            ;2715 ;
            ;2716 ;
            ;2717 ; SYNC POINT
            ;2718 ;
            ;2719 ;
            ;2720 ;
            ;2721 ;--
            ;2722 ;
            ;2723 ;*****************************************************************
            ;2724
            ;2725
            ;2726
            ;2727 ;/////////////////////////////////////////////////
            ;2728 ;*
            ;2729 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
            ;2730 ;-
            ;2731
            ;2732
            ;2733 =0
U 1070, 0018,0039,8980,09F8,0000,10BC  ;2734 MULT9: NEWTST[.D]
U 1071, 0003,003C,0180,0A80,0000,11BD  ;2735  R[0]_0    ; PATTERN COUNTER
U 11BD, 0818,0038,4D80,0800,0000,11BE  ;2736  D_K[.FF00]
U 11BE, 0819,0030,4180,0800,0000,11BF  ;2737  D_D.OR.K[.80]
U 11BF, 0001,0028,0180,0A88,0000,11C0  ;2738  R[1]_NOT.D   ; MASK FOR BITS <15:07>
U 11C0, 0000,003C,6180,0800,0084,71C1  ;2739  SC_K[.F]
U 11C1, 0000,003C,0580,0800,0084,B1C2  ;2740  SC_SC-K[.1]
U 11C2, 0801,0034,0180,0800,0000,11C3  ;2741  ALU_D.AND.MASK,D_ALU
U 11C3, 0001,0028,0180,0A90,0000,11C4  ;2742  R[2]_NOT.D   ; MASK FOR BITS <15,13:7>
U 11C4, 0818,0038,7980,0800,0000,11C5  ;2743  D_K[.30]
U 11C5, 0819,0030,8980,0800,0000,11C6  ;2744  D_D.OR.K[.D]
U 11C6, 0001,003C,0180,0990,0000,1072  ;2745  RC[2]_D    ; BIT COUNT OF 61
U 1072, 0000,003D,0180,0800,0000,10CE  ;2746 =0 ERLOOP
U 1073, 0810,0038,0180,0910,0082,11C7  ;2747 MUL17: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 11C7, 0019,0034,7580,0800,0010,11C8  ;2748  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING MULTIPLICAND
            ;2749       ; LO WITH PATTERN, MULTIPLICAND HI
            ;2750       ; WITH ZERO
U 11C8, 0000,013C,0180,0800,0000,1074  ;2751  Z?
U 1074, 0000,003C,0180,0800,0000,11CB  ;2752 =0 J/MUL18
U 1075, 0803,001C,0180,0800,0000,11C9  ;2753  D_NOT.MASK   ; FLOATING 1
U 11C9, 0019,0024,5980,0AA8,0000,11CA  ;2754  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT BITS <6:0>
U 11CA, 0003,003C,0180,0AA0,0000,11CD  ;2755  R[4]_0,J/MUL19
U 11CB, 0003,003C,0180,0AA8,0000,11CC  ;2756 MUL18: R[5]_0    ; MULTIPLICAND LO GETS 0
U 11CC, 0003,001C,0180,0AA0,0000,11CD  ;2757  R[4]_NOT.MASK   ; FLOATING 1
            ;2758
            ;2759 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
            ;2760
U 11CD, 0800,003C,0180,0A20,0000,11CE  ;2761 MUL19: D_R[4]
```

```
U 11CE. 0001,003C,0180,09B0,0000,11CF  ;2762  RC[6]_D   ; SAVE HI
U 11CF. 0800,003C,0180,0A28,0000,11D0  ;2763  D_R[5]
U 11D0. 0001,003C,0180,09A8,0000,11D1  ;2764  RC[5]_D   ; SAVE LO
U 11D1. 0000,003C,01C0,0A20,0000,11D2  ;2765  Q_R[4]
U 11D2. 0800,003C,0180,0A28,0000,11D3  ;2766  D_R[5]
U 11D3. 0600,003C,0080,0800,0000,11D4  ;2767  D_D.RIGHT,SI/1
U 11D4. 0000,003C,01B0,0800,0000,11D5  ;2768  Q_Q.RIGHT,SI/ZERO
U 11D5. 0001,203C,0180,0AA0,0000,11D6  ;2769  R[4]_Q
U 11D6. 0001,003C,0180,0AA8,0000,1076  ;2770  R[5]_D
          :2771
          :2772 ; PACK THE DATA
          :2773
U 1076. 0000,003D,0180,0800,0000,1137  ;2774 =0 CALL[UNSEQA]  ; PACK UNNORMALIZED EXPECTED PRODUCT
U 1077. 0800,003C,0180,0A30,0000,11D7  ;2775  D_R[6]
U 11D7. 080D,0034,0180,0A10,0000,11D8  ;2776  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 11D8. 0001,003C,0180,0AB0,0000,11D9  ;2777  R[6]_D
U 11D9. 0819,0030,3180,0800,0000,11DA  ;2778  D_D.OR.K[.40]  ; SET BIT 6
U 11DA. 0001,003C,0180,09D0,0000,11DB  ;2779  RC[0A]_D  ; UNNORMALIZED EXPECTED PRODUCT HI
U 11DB. 0800,003C,0180,0A38,0000,11DC  ;2780  D_R[7]
U 11DC. 0001,003C,0180,09C0,0000,11DD  ;2781  RC[8]_D   ; UNNORMALIZED EXPECTED PRODUCT LO
          :2782
          :2783 ; GENERATE NORMALIZED EXPECTED PRODUCT
          :2784
U 11DD. 0810,0038,0180,0930,0000,11DE  ;2785  D_RC[6]   ; RESTORE HI
U 11DE. 0001,003C,0180,0AA0,0000,11DF  ;2786  R[4]_D
U 11DF. 0810,0038,0180,0928,0000,11E0  ;2787  D_RC[5]   ; RESTORE LO
U 11E0. 0018,0038,31C0,0800,0000,11E1  ;2788  Q_K[.40]
U 11E1. 001D,2014,0180,0AA8,0010,11E2  ;2789  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 11E2. 0000,1B3C,0180,0800,0000,103E  ;2790  ALU.C?
U 103E. 0000,003C,0180,0800,0000,1078  ;2791 =1110 J/PAK9
U 103F. 0800,003C,0180,0A20,0000,11E3  ;2792  D_R[4]
U 11E3. 0019,0014,0580,0AA0,0000,1078  ;2793  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
          :2794
          :2795 ; PACK NORMALIZED EXPECTED PRODUCT
          :2796
          :2797 =0
U 1078. 0000,003D,0180,0800,0000,1137  ;2798 PAK9: CALL[UNSEQA]  ; PACK NORMALIZED EXPECTED PRODUCT
U 1079. 0800,003C,0180,0A30,0000,11E4  ;2799  D_R[6]   ;
U 11E4. 080D,0034,0180,0A08,0000,11E5  ;2800  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11E5. 0001,003C,0180,0AB0,0000,11E6  ;2801  R[6]_D
U 11E6. 0001,003C,0180,09F0,0000,11E7  ;2802  RC[0E]_D  ; SAVE NORMALIZED EXPECTED PRODUCT HI
U 11E7. 0800,003C,0180,0A38,0000,11E8  ;2803  D_R[7]
U 11E8. 0001,003C,0180,09E0,0000,11E9  ;2804  RC[0C]_D  ; SAVE NORMALIZED EXPECTED PRODUCT LO
          :2805
          :2806 ; PACK OPERANDS FOR MULTIPLY OPERATION
          :2807
U 11E9. 0810,0038,0180,0930,0000,11EA  ;2808  D_RC[6]
U 11EA. 0001,003C,0180,0AA0,0000,11EB  ;2809  R[4]_D   ; RESTORE  HI
U 11EB. 0810,0038,0180,0928,0000,11EC  ;2810  D_RC[5]
U 11EC. 0001,003C,0180,0AA8,0000,107A  ;2811  R[5]_D   ; RESTORE  LO
U 107A. 0000,003D,0180,0800,0000,1137  ;2812 =0 CALL[UNSEQA]
U 107B. 0800,003C,0180,0A30,0000,11ED  ;2813  D_R[6]   ;
U 11ED. 080D,0034,0180,0A08,0000,11EE  ;2814  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 11EE. 0001,003C,0180,0AB0,0000,11EF  ;2815  R[6]_D   ; MULTIPLICAND HI MASKED
U 11EF. 0003,003C,0180,0AC0,0000,11F0  ;2816  R[8]_0   ; MULTIPLIER HI
```

```
U 11F0. 0001.003C.0180.09A0.0000.11F1  ;2817  RC[4]_D   ; SAVE MULTIPLICAND HI
U 11F1. 0003.003C.0180.0AC8.0000.11F2  ;2818  R[9]_Q   ; MULTIPLIER LO
U 11F2. 0800.003C.0180.0A38.0000.11F3  ;2819  D_R[7]
U 11F3. 0001.003C.0180.0998.0000.107C  ;2820  RC[3]_D   ; SAVE MULTIPLICAND LO
          ;2821
          ;2822 ; LOAD OPERANDS, MULTIPLY
          ;2823
U 107C. 0000.003D.0180.0800.0000.1014  ;2824 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
          ;2825
          ;2826
          ;2827
U 107D. 0000.003C.0180.0800.0000.107E  ;2828  NOP
U 107E. 0000.003D.0180.0800.0000.101C  ;2829 =0 CALL[MULD1]  ; MULTIPLY DOUBLE
U 107F. 0001.203C.0180.09D8.0000.11F4  ;2830  RC[0B]_Q  ; SAVE NORMALIZED PRODUCT LO
U 11F4. 081C.2034.0180.0A08.0000.11F5  ;2831  D_D.AND.R[1]  ; MASK OUT BITS <15:7>
U 11F5. 0001.003C.0180.09E8.0000.11F6  ;2832  RC[0D]_D  ; SAVE NORMALIZED PRODUCT HI
U 11F6. 0810.0038.0180.0908.0000.11F7  ;2833  D_RC[1]
U 11F7. 080D.0034.0180.0A10.0000.11F8  ;2834  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15:7>
U 11F8. 0001.003C.0180.09C8.0000.11F9  ;2835  RC[9]_D   ; SAVE UNNORMALIZED PRODUCT HI
U 11F9. 0000.003C.0180.0900.0000.11FA  ;2836  LC_RC[0]
U 11FA. 0010.0038.0180.09B8.0000.11FB  ;2837  RC[7]_LC  ; SAVE UNNORMALIZED PRODUCT LO
U 11FB. 0011.2020.0180.0960.0010.11FC  ;2838  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED PRODUCT LO
U 11FC. 0000.013C.0180.0800.0000.1080  ;2839  Z?
U 1080. 0000.003D.0180.0800.0000.10CF  ;2840 =0 ERROR1
U 1081. 0010.0038.01C0.0968.0000.11FD  ;2841  Q_RC[0D]
U 11FD. 0011.2020.0180.0970.0010.11FE  ;2842  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED PRODUCT HI
U 11FE. 0000.013C.0180.0800.0000.1082  ;2843  Z?
U 1082. 0000.003D.0180.0800.0000.10CF  ;2844 =0 ERROR1
U 1083. 0010.0038.01C0.0940.0000.11FF  ;2845  Q_RC[8]
U 11FF. 0011.2020.0180.0938.0010.1200  ;2846  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1200. 0000.013C.0180.0800.0000.1084  ;2847  Z?
U 1084. 0000.003D.0180.0800.0000.10CF  ;2848 =0 ERROR1
U 1085. 0010.0038.01C0.0950.0000.1201  ;2849  Q_RC[0A]
U 1201. 0011.2020.0180.0948.0010.1202  ;2850  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 1202. 0000.013C.0180.0800.0000.1086  ;2851  Z?
U 1086. 0000.003D.0180.0800.0000.10CF  ;2852 =0 ERROR1
U 1087. 0810.0038.0180.0910.0010.1203  ;2853  ALU_RC[2],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 1203. 0000.013C.0180.0800.0000.1088  ;2854  Z?
U 1088. 0019.0000.0580.0990.0000.1073  ;2855 =0 ALU_D-K[.1],RC[2]_ALU,J/MUL17 ; DECREMENT BIT COUNTER
U 1089. 0000.003C.0180.0800.0000.108A  ;2856  NOP
          ;2857
```

```
;2858  .PAGE "TEST 2A4 MULTIPLY DOUBLE, 0.1  BY FLOATING ONE"
;2859 ;*****************************************************************
;2860 ;**
;2861 ;
;2862 ; MULTIPLY DOUBLE, 0.1  BY FLOATING ONE
;2863 ;
;2864 ;
;2865 ; TEST DESCRIPTION
;2866 ;
;2867 ; THIS TEST SUCCESSIVELY MULTIPLIES A FLOATING (1) PATTERN
;2868 ; IN THE MULTIPLIER REGISTER, WITH THE MULTIPLICAND REGISTER
;2869 ; CONTAINING ONLY THE HIDDEN BIT SET.
;2870 ;
;2871 ;
;2872 ; TEST SEQUENCE   DESCRIPTION
;2873 ; 1  GENERATE OPERANDS AND EXPECTED RESULTS
;2874 ; 2  PACK THE DATA
;2875 ; 3  PERFORM THE MULTIPLY
;2876 ; 4  VERIFY RESULTS
;2877 ;
;2878 ; LOCATION CONTENTS
;2879 ;
;2880 ; FPA UCODE USED
;2881 ;
;2882 ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;2883 ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;2884 ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;2885 ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;2886 ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;2887 ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;2888 ; 028: NAD/028
;2889 ; 0E6: FADC/A,BSC/FAL.HL,OPLD/MC,NAD/0E7
;2890 ; 0E7: FSDC/B,BSC/FAL.LH,OPLD/MP,NAD/0C8
;2891 ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;2892 ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;2893 ; 0CA: NAD/028
;2894 ; 028: NAD/028
;2895 ; 0D0: BEN6,NAD/0D0,BMXC/LB,BSC,PQ
;2896 ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;2897 ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;2898 ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;2899 ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;2900 ;
;2901 ;
;2902 ; LOGIC DESCRIPTION
;2903 ;
;2904 ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;2905 ; LOGIC ARE ON THE FMH AND FML LOGIC MODULES.
;2906 ;
;2907 ; ERROR DESCRIPTION
;2908 ;
;2909 ; PACKED EXPECTED NORMALIZED PRODUCT HI
;2910 ; PACKED RECEIVED NORMALIZED PRODUCT HI
;2911 ; PACKED EXPECTED NORMALIZED PRODUCT LO
;2912 ; PACKED RECEIVED NORMALIZED PRODUCT LO
```

```
                ;2913 ; PACKED EXPECTED UNNORMALIZED PRODUCT HI
                ;2914 ; PACKED RECEIVED UNNORMALIZED PRODUCT HI
                ;2915 ; EXPECTED UNNORMALIZED PRODUCT LO
                ;2916 ; RECEIVED UNNORMALIZED PRODUCT LO
                ;2917 ; MULTIPLICAND REGISTER HI
                ;2918 ; MULTIPLICAND REGISTER LO
                ;2919 ; MULTIPLIER REGISTER HI
                ;2920 ; MULTIPLIER REGISTER LO
                ;2921 ; BIT COUNTER
                ;2922 ;
                ;2923 ;
                ;2924 ;
                ;2925 ; SYNC POINT
                ;2926 ;
                ;2927 ;
                ;2928 ;
                ;2929 ;--
                ;2930 ;
                ;2931 ;*****************************************************************
                ;2932
                ;2933
                ;2934
                ;2935 ;/////////////////////////////////////////////////
                ;2936 ;*
                ;2937 ; FLOAT A 1 THROUGH THE MULTIPLICAND WITH MULTIPLIER=0.1.
                ;2938 ;-
                ;2939
                ;2940
                ;2941 =0
U 108A. 0018.0039.B9B0.09F8.0000.10BC   ;2942 MULT10: NEWTST[.D]
U 108B. 0818.0038.4D80.0800.0000.1204   ;2943  D_K[.FF00]
U 1204. 0819.0030.4180.0800.0000.1205   ;2944  D_D.OR.K[.80]
U 1205. 0001.0028.0180.0A88.0000.1206   ;2945  R[1]_NOT.D  ; MASK FOR BITS <15:07>
U 1206. 000D.003C.6180.0800.0084.7207   ;2946  SC_K[.F]
U 1207. 000D.003C.0580.0800.0084.B208   ;2947  SC_SC-K[.1]
U 1208. 0801.0034.0180.0800.0000.1209   ;2948  ALU_D.AND.MASK,D_ALU
U 1209. 0001.0028.0180.0A90.0000.120A   ;2949  R[2]_NOT.D  ; MASK FOR BITS <15,13:7>
U 120A. 0818.0038.7980.0800.0000.120B   ;2950  D_K[.30]
U 120B. 0819.0030.8980.0800.0000.120C   ;2951  D_D.OR.K[.D]
U 120C. 0001.003C.0180.0990.0000.108C   ;2952  RC[2]_D   ; BIT COUNT OF 61
U 108C. 000D.003D.0180.0800.0000.10CE   ;2953 =0 ERLOOP
U 108D. 0810.0038.0180.0910.0082.120D   ;2954 MUL20: D_RC[2],SC_RC[2] ; D GETS  COUNT, SC GETS PATTERN
U 120D. 0019.0034.7580.0800.0010.120E   ;2955  ALU_D.AND.K[.20],CLK.UBCC ; IF AT BIT 31, START LOADING MULTIPLIER
                ;2956       ; LO WITH PATTERN, MULTIPLIER HI
                ;2957       ; WITH ZERO.
U 120E. 000D.013C.0180.0800.0000.108E   ;2958  Z?
U 108E. 000D.003C.0180.0800.0000.1211   ;2959 =0 J/MUL21
U 108F. 0803.001C.0180.0800.0000.120F   ;2960  D_NOT.MASK  ; FLOATING 1
U 120F. 0019.0024.5980.0AA8.0000.1210   ;2961  ALU_D[ANDNOT]K[.7F],R[5]_ALU ; MASK OUT BITS <6:0>
U 1210. 0003.003C.0180.0AA0.0000.1213   ;2962  R[4]_0,J/MUL22
U 1211. 0003.003C.0180.0AA8.0000.1212   ;2963 MUL21: R[5]_0   ; MULTIPLIER LO GETS 0
U 1212. 0003.001C.0180.0AA0.0000.1213   ;2964  R[4]_NOT.MASK  ; FLOATING 1
                ;2965
                ;2966 ; GENERATE UNNORMALIZED EXPECTED PRODUCT
                ;2967
```

```
ZZ-ESKAR-V22  TEST 2A4 MULTIPLY DOUBLE, 0.1  BY FLOATING ONE                                  SEQ 2851
ESKAR66.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40                        Page   7

U 1213. 0800.003C.0180.0A20.0000.1214  ;2968 MUL22: D_R[4]
U 1214. 0001.003C.0180.09A0.0000.1215  ;2969  RC[4]_D   ; SAVE HI
U 1215. 0800.003C.0180.0A28.0000.1216  ;2970  D_R[5]
U 1216. 0001.003C.0180.0998.0000.1217  ;2971  RC[3]_D   ; SAVE LO
U 1217. 0000.003C.01C0.0A20.0000.1218  ;2972  Q_R[4]
U 1218. 0800.003C.0180.0A28.0000.1219  ;2973  D_R[5]
U 1219. 0600.003C.0080.0800.0000.121A  ;2974  D_D.RIGHT,SI/1
U 121A. 0000.003C.01B0.0800.0000.121B  ;2975  Q_Q.RIGHT,SI/ZERO
U 121B. 0001.203C.0180.0AA0.0000.121C  ;2976  R[4]_Q
U 121C. 0001.003C.0180.0AA8.0000.1090  ;2977  R[5]_D
           ;2978
           ;2979 ; PACK THE DATA
           ;2980
U 1090. 0000.003D.0180.0800.0000.1137  ;2981 =0 CALL[UNSEQA]  ; PACK UNNORMALIZED EXPECTED PRODUCT
U 1091. 0800.003C.0180.0A30.0000.121D  ;2982  D_R[6]
U 121D. 080D.0034.0180.0A10.0000.121E  ;2983  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 121E. 0001.003C.0180.0AB0.0000.121F  ;2984  R[6]_D
U 121F. 0819.0030.3180.0800.0000.1220  ;2985  D_D.OR.K[.40]  ; SET BIT 6
U 1220. 0001.003C.0180.09D0.0000.1221  ;2986  RC[0A]_D  ; SAVE
U 1221. 0800.003C.0180.0A38.0000.1222  ;2987  D_R[7]
U 1222. 0001.003C.0180.09C0.0000.1223  ;2988  RC[8]_D   ; SAVE
           ;2989
           ;2990 ; GENERATE NORMALIZED EXPECTED PRODUCT
           ;2991
U 1223. 0810.0038.0180.0920.0000.1224  ;2992  D_RC[4]   ; RESTORE HI
U 1224. 0001.003C.0180.0AA0.0000.1225  ;2993  R[4]_D
U 1225. 0810.0038.0180.0918.0000.1226  ;2994  D_RC[3]   ; RESTORE LO
U 1226. 0018.0038.31C0.0800.0000.1227  ;2995  Q_K[.40]
U 1227. 001D.2014.0180.0AA8.0010.1228  ;2996  ALU_Q+D,CLK.UBCC,R[5]_ALU
U 1228. 0000.1B3C.0180.0800.0000.106E  ;2997  ALU.C?
U 106E. 0000.003C.0180.0800.0000.1092  ;2998 =1110 J/PAK10
U 106F. 0800.003C.0180.0A20.0000.1229  ;2999  D_R[4]
U 1229. 0019.0014.0580.0AA0.0000.1092  ;3000  ALU_D+K[.1],R[4]_ALU ; INCREMENT HI IF CARRY OUT
           ;3001
           ;3002 ; PACK NORMALIZED EXPECTED PRODUCT
           ;3003
           ;3004 =0
U 1092. 0000.003D.0180.0800.0000.1137  ;3005 PAK10: CALL[UNSEQA]  ; PACK NORMALIZED EXPECTED PRODUCT
U 1093. 0800.003C.0180.0A30.0000.122A  ;3006  D_R[6]
U 122A. 080D.0034.0180.0A08.0000.122B  ;3007  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 122B. 0001.003C.0180.0AB0.0000.122C  ;3008  R[6]_D
U 122C. 0001.003C.0180.09F0.0000.122D  ;3009  RC[0E]_D  ; SAVE NORMALIZED EXPECTED PRODUCT HI
U 122D. 0800.003C.0180.0A38.0000.122E  ;3010  D_R[7]
U 122E. 0001.003C.0180.09E0.0000.122F  ;3011  RC[0C]_D  ; SAVE NORMALIZED EXPECTED PRODUCT LO
           ;3012
           ;3013 ; PACK OPERANDS FOR MULTIPLY OPERATION
           ;3014
U 122F. 0810.0038.0180.0920.0000.1230  ;3015  D_RC[4]
U 1230. 0001.003C.0180.0AA0.0000.1231  ;3016  R[4]_D   ; RESTORE  HI
U 1231. 0810.0038.0180.0918.0000.1232  ;3017  D_RC[3]
U 1232. 0001.003C.0180.0AA8.0000.1094  ;3018  R[5]_D   ; RESTORE  LO
U 1094. 0000.003D.0180.0800.0000.1137  ;3019 =0 CALL[UNSEQA]
U 1095. 0800.003C.0180.0A30.0000.1233  ;3020  D_R[6]
U 1233. 080D.0034.0180.0A08.0000.1234  ;3021  ALU_D[AND]R[1],D_ALU ; MASK OUT BITS <15:7>
U 1234. 0001.003C.0180.0AC0.0000.1235  ;3022  R[8]_D   ; MULTIPLIER HI MASKED
```

```
U 1235. 0001,003C,0180,09B0,0000,1236  ;3023  RC[6]_D   ; SAVE MULTIPLIER HI
U 1236. 0003,003C,0180,0AB0,0000,1237  ;3024  R[6]_Q   ; MULTIPLICAND HI
U 1237. 0800,003C,0180,0A38,0000,1238  ;3025  D_R[7]
U 1238. 0001,003C,0180,0AC8,0000,1239  ;3026  R[9]_D   ; MULTIPLIER LO
U 1239. 0001,003C,0180,09A8,0000,123A  ;3027  RC[5]_D   ; SAVE MULTIPLIER LO
U 123A. 0003,003C,0180,0AB8,0000,1096  ;3028  R[7]_Q   ; MULTIPLIER LO
          ;3029
          ;3030 ; LOAD OPERANDS, MULTIPLY
          ;3031
U 1096. 0000,003D,0180,0800,0000,1014  ;3032 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
          ;3033
U 1097. 0000,003C,0180,0800,0000,1098  ;3034  NOP
U 1098. 0000,003D,0180,0800,0000,101C  ;3035 =0 CALL[MULD1]  ; MULTIPLY DOUBLE
U 1099. 0001,203C,0180,09D8,0000,123B  ;3036  RC[0B]_Q  ; SAVE NORMALIZED PRODUCT LO
U 123B. 081C,2034,0180,0A08,0000,123C  ;3037  D_D.AND.R[1]  ; MASK OUT BITS<15:07>
U 123C. 0001,003C,0180,09E8,0000,123D  ;3038  RC[0D]_D  ; SAVE NORMALIZED PRODUCT HI
U 123D. 0810,0038,0180,0908,0000,123E  ;3039  D_RC[1]
U 123E. 080D,0034,0180,0A10,0000,123F  ;3040  ALU_D[AND]R[2],D_ALU ; MASK OUT BITS <15,13:7>
U 123F. 0001,003C,0180,09C8,0000,1240  ;3041  RC[9]_D   ; SAVE UNNORMALIZED PRODUCT HI
U 1240. 0000,003C,0180,0900,0000,1241  ;3042  LC_RC[0]
U 1241. 0010,0038,0180,09B8,0000,1242  ;3043  RC[7]_LC  ; SAVE UNNORMALIZED PRODUCT LO
U 1242. 0011,2020,0180,0960,0010,1243  ;3044  ALU_Q.XOR.RC[0C],CLK.UBCC ; CHECK NORMALIZED PRODUCT LO
U 1243. 0000,013C,0180,0800,0000,109A  ;3045  Z?
U 109A. 0000,003D,0180,0800,0000,10CF  ;3046 =0 ERROR1
U 109B. 0010,0038,01C0,0968,0000,1244  ;3047  Q_RC[0D]
U 1244. 0011,2020,0180,0970,0010,1245  ;3048  ALU_Q.XOR.RC[0E],CLK.UBCC ; CHECK NORMALIZED PRODUCT HI
U 1245. 0000,013C,0180,0800,0000,109C  ;3049  Z?
U 109C. 0000,003D,0180,0800,0000,10CF  ;3050 =0 ERROR1
U 109D. 0010,0038,01C0,0940,0000,1246  ;3051  Q_RC[8]
U 1246. 0011,2020,0180,0938,0010,1247  ;3052  ALU_Q.XOR.RC[7],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT LO
U 1247. 0000,013C,0180,0800,0000,109E  ;3053  Z?
U 109E. 0000,003D,0180,0800,0000,10CF  ;3054 =0 ERROR1
U 109F. 0010,0038,01C0,0950,0000,1248  ;3055  Q_RC[0A]
U 1248. 0011,2020,0180,0948,0010,1249  ;3056  ALU_Q.XOR.RC[9],CLK.UBCC ; CHECK UNNORMALIZED PRODUCT HI
U 1249. 0000,013C,0180,0800,0000,10A0  ;3057  Z?
U 10A0. 0000,003D,0180,0800,0000,10CF  ;3058 =0 ERROR1
U 10A1. 0810,0038,0180,0910,0010,124A  ;3059  ALU_RC[2],CLK.UBCC,D_ALU ; CHECK BIT COUNTER
U 124A. 0000,013C,0180,0800,0000,10A2  ;3060  Z?
U 10A2. 0019,0000,0580,0990,0000,108D  ;3061 =0 ALU_D-K[.1],RC[2]_ALU,J/MUL20 ; DECREMENT BIT COUNTER
U 10A3. 0000,003C,0180,0800,0000,10A4  ;3062  NOP
          ;3063
          ;3064
          ;3065
          ;3066
          ;3067
          ;3068
          ;3069
          ;3070
```

```
;3071   .PAGE "TEST 2A5 VERIFY THE MULTIPLIER ROMS"
;3072  ;*****************************************************************
;3073  ;**
;3074  ;
;3075  ; VERIFY THE MULTIPLIER ROMS
;3076  ;
;3077  ; TEST DESCRIPTION
;3078  ;
;3079  ; THIS TEST PERFORMS A MULTIPLY DOUBLE WITH OPERANDS
;3080  ; THAT WILL COMPLETELY VERIFY THE MULTILPIER ROMS.
;3081  ;
;3082  ; TEST SEQUENCE   DESCRIPTION
;3083  ; 1  GENERATE MULTIPLICAND
;3084  ; 2  FETCH THE MULTIPLIER
;3085  ; 3  PERFORM THE MULTIPLY
;3086  ; 4  VERIFY RESULTS
;3087  ;
;3088  ; FPA UCODE USED
;3089  ;
;3090  ; LOCATION CONTENTS
;3091  ; 068: BSC/R,SCR/CPU,EAC/LDA,SGNC/LDSA,WAIT,NAD/069
;3092  ; 069: BSC/R,SCR/CPU,EAC/LDB,SGNC/LDSB,WAIT,NAD/06A
;3093  ; 06A: BSC/R,SCR/CPU,FADC/AR,WAIT,NAD/06C
;3094  ; 06C: BSC/R,SCR/CPU,FADC/AR0,WAIT,NAD/06D
;3095  ; 06D: BSC/R,SCR/CPU,FADC/BR,WAIT,NAD/083
;3096  ; 083: BSC/R,SCR/CPU,FADC/BR0,WAIT,NAD/028
;3097  ; 028: NAD/028
;3098  ; 0C3: FADC/A,BSC/FAL.LH,OPLD/MC.P0,NAD/0C4
;3099  ; 0C4: FADC/A,BSC/FAL.LH,OPLD/MP1,NAD/0C5
;3100  ; 0C5: FADC/B,BSC/FAL.HL,OPLD/MC0,NAD/0C6
;3101  ; 0C6: FADC/B,BSC/FAL.HL,OPLD/MC1,NAD/0C8
;3102  ; 0C8: MCTL/CNT,AMXC/LA,BMXC/LB,EALUC/A+B,EAC/LDPR,NAD/0C9
;3103  ; 0C9: AMXC/PR,BMXC/#80,EALUC/A-B,EAC/LDPR,NAD/0CA
;3104  ; 0CA: NAD/028
;3105  ; 028: NAD/028
;3106  ; 0D0: BEN/6,NAD/0D0,BMXC/LB,BSC/PQ
;3107  ; 0D1: BSC/PQ,FADC/AR,NRC/LD,MSC/RR.0,WAIT,FPSYNC,NAD/0D2
;3108  ; 0D2: FADC/A,BSC/FAL.LH,NAD/0D3,WAIT
;3109  ; 0D3: BSC/NL,OPLD/INIT,AMXC/LA,EALUC/A,NAD/0D4
;3110  ; 0D4: BSC/NH,AMXC/LA,EALUC/A,NAD/PSTALL
;3111  ;
;3112  ; LOGIC DESCRIPTION
;3113  ;
;3114  ; THE MULTIPLIER OPERAND REGISTERS AND THE MULTIPLIER
;3115  ; LOGIC ARE ON THE FMH AND FML LOGIC MODULES.
;3116  ;
;3117  ; ERROR DESCRIPTION
;3118  ;
;3119  ; PACKED MULTIPLIER HI
;3120  ; PACKED MULTIPLIER LO
;3121  ; MULTIPLICAND HI
;3122  ; MULTIPLICAND LO
;3123  ; EXPECTED NORMALIZED PRODUCT HI
;3124  ; RECEIVED NORMALIZED PRODUCT HI
;3125  ; EXPECTED NORMALIZED PRODUCT LO
```

```
            ;3126 ; RECEIVED NORMALIZED PRODUCT LO
            ;3127 ; RECEIVED UNNORMALIZED PRODUCT HI
            ;3128 ; RECEIVED UNNORMALIZED PRODUCT LO
            ;3129 ; MASK
            ;3130 ;
            ;3131 ;
            ;3132 ;
            ;3133 ; SYNC POINT
            ;3134 ;
            ;3135 ;
            ;3136 ;
            ;3137 ;--
            ;3138 ;
            ;3139 ;*****************************************************************
            ;3140
            ;3141
            ;3142 ; MULTIPLY DOUBLE WITH SELECTED OPERANDS
            ;3143
            ;3144 =0
U 10A4, 0018.0039.8980.09F8.0000.10BC  ;3145 MULT11: NEWTST[.D]
U 10A5, 0818.0038.4D80.0800.0000.124B  ;3146  D_K[.FF00]
U 124B, 0819.0030.4180.0800.0000.124C  ;3147  D_D.OR.K[.80]
U 124C, 0001.0028.0180.09A0.0000.124D  ;3148  RC[4]_NOT.D  ; MASK OUT BITS <15:7>
U 124D, 0003.003C.0180.0A90.0000.124E  ;3149  R[2]_0   ; MULTIPLICAND COUNTER
U 124E, 0818.0038.4980.0800.0000.124F  ;3150  D_K[.FF]
U 124F, 0819.0000.0580.0800.0000.1250  ;3151  D_D-K[.1]
U 1250, 0500.003C.0180.0800.0000.1251  ;3152  D_D.LEFT  ; MULTIPLIER  WORD COUNT ADDRESS 1FC
U 1251, 0001.003C.0180.0800.0200.1252  ;3153  VA_D
U 1252, 0000.003C.0180.4000.0000.1253  ;3154  D[LONG]_CACHE
U 1253, 0001.003C.0180.0AE8.0000.1254  ;3155  R[0D]_D   ; SAVE MULTIPLIER  WORD COUNT
U 1254, 0818.0038.7580.0800.0000.1255  ;3156  D_K[.20]
U 1255, 0100.003C.0180.0800.0000.1256  ;3157  D_D.LEFT2
U 1256, 0100.003C.0180.0800.0000.1257  ;3158  D_D.LEFT2
U 1257, 0001.003C.0180.0AD8.0000.1258  ;3159  R[0B]_D   ; FIRST MULTIPLIER  ADDRESS IS 200
U 1258, 0818.0038.C980.0800.0000.1259  ;3160  D_K[.3030]
U 1259, 0200.003C.0180.0800.0000.125A  ;3161  D_D.RIGHT2
U 125A, 0200.003C.0180.0800.0000.125B  ;3162  D_D.RIGHT2
U 125B, 0019.0024.6180.0AD0.0000.10A6  ;3163  R[0A]_ALU,ALU_D.ANDNOT.K[.F] ; FIRST PRODUCT ADDRESS IS 300
U 10A6, 0000.003D.0180.0800.0000.10CE  ;3164 =0 ERLOOP
            ;3165
            ;3166 ; FETCH MULTIPLIER
            ;3167
U 10A7, 0800.003C.0180.0A58.0000.125C  ;3168 MULD: D_R[0B]  ; SAVE
U 125C, 0001.003C.0180.0800.0200.125D  ;3169  VA_D   ; ADDRESS
U 125D, 0000.003C.0180.4C00.0000.125E  ;3170  D[LONG]_CACHE
U 125E, 0811.0034.0180.0920.0000.125F  ;3171  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS, <15:7>
U 125F, 0001.003C.0180.09F0.0000.1260  ;3172  RC[0E]_D  ; SAVE MULTIPLIER HI
U 1260, 0000.003C.0180.0803.0000.1261  ;3173  VA_VA+4   ; ADDRESS
U 1261, 0000.003C.0180.4000.0000.1262  ;3174  D[LONG]_CACHE
U 1262, 0001.003C.0180.09E8.0000.10A8  ;3175  RC[0D]_D  ; SAVE MULTIPLIER LO
            ;3176
            ;3177 ; GENERATE MULTIPLICAND
            ;3178
U 10A8, 0000.003D.0180.0800.0000.1158  ;3179 =0 CALL[MCAND]
U 10A9, 0811.0034.0180.0920.0000.1263  ;3180  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
```

```
U 1263. 0001.003C.0180.0AC0.0000.1264  ;3181  R[8]_D   ; HOLD MULTIPLICAND HI FOR MULTIPLY
U 1264. 0001.003C.0180.09E0.0000.1265  ;3182  RC[0C]_D  ; SAVE
U 1265. 0001.203C.0180.0AC8.0000.1266  ;3183  R[9]_Q   ; HOLD MULTIPLICAND LO FOR MULTIPLY
U 1266. 0001.203C.0180.09D8.0000.1267  ;3184  RC[0B]_Q  ; SAVE
U 1267. 0810.0038.0180.0970.0000.1268  ;3185  D_RC[0E]
U 1268. 0001.003C.0180.0AB0.0000.1269  ;3186  R[6]_D   ; HOLD MULTIPLIER HI FOR MULTIPLY
U 1269. 0810.0038.0180.0968.0000.126A  ;3187  D_RC[0D]  ;
U 126A. 0001.003C.0180.0AB6.0000.10AA  ;3188  R[7]_D   ; HOLD MULTIPLIER LO FOR MULTIPLY
          ;3189
          ;3190 ; PERFORM THE MULTIPLY
          ;3191
U 10AA. 0000.003D.0180.0800.0000.1014  ;3192 =0 CALL[LD.MUL]  ; LOAD OPERANDS INTO FAD AR,BR LATCHES.
          ;3193
U 10AB. 0000.003C.0180.0800.0000.10AC  ;3194  NOP
U 10AC. 0000.003D.0180.0800.0000.1022  ;3195 =0 CALL[MULD2]  ; MULTIPLY DOUBLE
          ;3196      ; READ DATA INTO Q AND D
U 10AD. 0001.203C.0180.09B8.0000.126B  ;3197  RC[7]_Q   ; SAVE NORMALIZED PRODUCT HI
U 126B. 0811.0034.0180.0920.0000.126C  ;3198  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 126C. 0001.003C.0180.09C8.0000.126D  ;3199  RC[9]_D   ; SAVE NORMALIZED PRODUCT LO
U 126D. 0810.0038.0180.0908.0000.126E  ;3200  D_RC[1]
U 126E. 0001.003C.0180.09B0.0000.126F  ;3201  RC[6]_D   ; SAVE UNNORMALIZED PRODUCT HI
U 126F. 0810.0038.0180.0900.0000.1270  ;3202  D_RC[0]
U 1270. 0001.003C.0180.09A8.0000.1271  ;3203  RC[5]_D   ; SAVE UNNORMALIZED PRODUCT LO
          ;3204
          ;3205 ; FETCH EXPECTED PRODUCT, VERIFY RESULTS.
          ;3206
U 1271. 0800.003C.0180.0A50.0000.1272  ;3207  D_R[0A]
U 1272. 0001.003C.0180.0800.0200.1273  ;3208  VA_D
U 1273. 0000.003C.0180.4000.0000.1274  ;3209  D[LONG]_CACHE  ; EXPECTED NORMALIZED PRODUCT HI
U 1274. 0811.0034.0180.0920.0000.1275  ;3210  ALU_D[AND]RC[4],D_ALU ; MASK OUT BITS <15:7>
U 1275. 0001.003C.0180.09D0.0000.1276  ;3211  RC[0A]_D  ; SAVE
U 1276. 0010.0038.01C0.0948.0000.1277  ;3212  Q_RC[9]
U 1277. 0011.2020.0180.0950.0010.1278  ;3213  ALU_Q.XOR.RC[0A],CLK.UBCC ; CHECK PROD. HI
U 1278. 0000.013C.0180.0800.0000.10AE  ;3214  Z?
U 10AE. 0000.003D.0180.0800.0000.10D1  ;3215 =0 ERROR2
U 10AF. 0000.003C.0180.0803.0000.1279  ;3216  VA_VA+4   ; ADDRESS
U 1279. 0000.003C.0180.4000.0000.127A  ;3217  D[LONG]_CACHE  ; EXPECTED NORMALIZED PRODUCT LO
U 127A. 0001.003C.0180.09C0.0000.127B  ;3218  RC[8]_D   ; SAVE
U 127B. 0010.0038.01C0.0938.0000.127C  ;3219  Q_RC[7]
U 127C. 0011.2020.0180.0940.0010.127D  ;3220  ALU_Q.XOR.RC[8],CLK.UBCC ; CHECK PROD. LO
U 127D. 0000.013C.0180.0800.0000.10B0  ;3221  Z?
U 10B0. 0000.003D.0180.0800.0000.10D1  ;3222 =0 ERROR2
U 10B1. 0000.003C.6180.0800.0000.127E  ;3223  KMX/.F
U 127E. 0018.0000.6180.0A10.0010.127F  ;3224  ALU_R[2]-K[.F],CLK.UBCC
U 127F. 0000.013C.0180.0800.0000.10B2  ;3225  Z?    ; 16 MULTIPLICANDS PROCESSED?
U 10B2. 0000.003C.0180.0800.0000.1280  ;3226 =0 J/MUL23   ; NO. UPDATE MULTIPLICAND AND PRODUCT
U 10B3. 0000.003C.0180.0800.0000.1284  ;3227  J/MUL24   ; YES. UPDATE PRODUCT, MULTIPLICAND
          ;3228      ; AND CHECK MULTIPLIER COUNTER
U 1280. 0818.0014.0580.0A10.0000.1281  ;3229 MUL23: ALU_R[2]+K[.1],D_ALU ; UPDATE TO NEXT MULTIPLICAND
U 1281. 0001.003C.0180.0A90.0000.1282  ;3230  R[2]_D
U 1282. 0818.0014.0180.0A50.0000.1283  ;3231  ALU_R[0A]+K[.8],D_ALU ; UPDATE TO NEXT PRODUCT
U 1283. 0001.003C.0180.0AD0.0000.10A7  ;3232  R[0A]_D,J/MULD
U 1284. 0003.003C.0180.0A90.0000.1285  ;3233 MUL24: R[2]_0   ; BACK TO FIRST MULTIPLICAND
U 1285. 0818.0014.0180.0A50.0000.1286  ;3234  ALU_R[0A]+K[.8],D_ALU  ; UP DATE TO NEXT PRODUCT
U 1286. 0001.003C.0180.0AD0.0000.1287  ;3235  R[0A]_D
```

```
U 1287. 0818.0000.0980.0A68.0010.1288  ;3236  ALU_R[0D]-K[.2],D_ALU,CLK.UBCC
U 1288. 0000.013C.0180.0800.0000.10B4  ;3237  Z?    ; 3 MULTIPLIERS PROCESSED?
U 10B4. 0001.003C.0180.0AE8.0000.1289  ;3238 =0 R[0D]_D,J/MUL25
U 10B5. 0000.003C.0180.0800.0000.128B  ;3239  J/ENTST    ; END OF TEST
U 1289. 0818.0014.0180.0A58.0000.128A  ;3240 MUL25: ALU_R[0B]+K[.8],D_ALU ; UPDATE MULTIPLIER ADDRESS
U 128A. 0001.003C.0180.0AD8.0000.10A7  ;3241  R[0B]_D,J/MULD
U 128B. 0000.003C.0180.0800.0000.10B6  ;3242 ENTST: NOP
            ;3243
            ;3244
            ;3245
            ;3246 ; OPCODE DATA
            ;3247 ;* 8
            ;3248 ; MULF R,R
            ;3249 ;$ 5544, 0056
            ;3250 ; MULL R,R
            ;3251 ;$ 55C4, 0056
            ;3252 ; MULD R,R
            ;3253 ;$ 5564, 0056
            ;3254 ; EMODF
            ;3255 ;$ 5554, 0056
            ;3256 ; EMODD
            ;3257 ;$ 5574, 0056
            ;3258
            ;3259
            ;3260
            ;3261 ; TABLE OF MULTIPLY PATTERNS
            ;3262
            ;3263 ; WORD COUNT MULTIPLIERS
            ;3264 ;* 1FC
            ;3265 ;$ 0006, 0000
            ;3266 ;* 200
            ;3267 ; MULTIPLIER PATTERN 1
            ;3268 ;$ 40FF, 11EE
            ;3269 ;$ BB88, 5522
            ;3270 ; MULTIPLIER PATTERN 2
            ;3271 ;$ 40EE, 00DD
            ;3272 ;$ AA77, 4411
            ;3273 ; MULTIPLIER PATTERN 3
            ;3274 ;$ 40DD, FFCC
            ;3275 ;$ 9966, 3300
            ;3276 ;
            ;3277 ; THE PATTERNS APPEARING IN THIS TABLE CONTAIN EXPONENT AND SIGN
            ;3278 ; THE PATTERNS IN THE ERROR DESCRIPTION HAVE THEIR EXPONENT AND
            ;3279 ; SIGN BITS MASKED OUT. THIS TEST MAKES USE OF THE FRACTION PORTION
            ;3280 ; OF THE PACKED DOUBLE FLOATING POINT DATA ONLY.
            ;3281 ;
            ;3282 ;* 300
            ;3283 ; EXPECTED PRODUCT PATTERN GROUP 1
            ;3284 ;$ 407F, 11EE
            ;3285 ;$ BB88, 5522
            ;3286 ;$ 4090, 8A29
            ;3287 ;$ 6A44, B8C6
            ;3288 ;$ 40A1, 8B5B
            ;3289 ;$ 76C5, 46FC
            ;3290 ;$ 40B2, 8C8D
```

```
;3291 ;$ 8345. D531
;3292 ;$ 40C3. 8DBF
;3293 ;$ 8FC6. 6367
;3294 ;$ 40D4. 8EF1
;3295 ;$ 9C46. F19C
;3296 ;$ 40E5. 9023
;3297 ;$ A8C7. 7FD1
;3298 ;$ 40F6. 9155
;3299 ;$ B548. 0E07
;3300 ;$ 4108. 0990
;3301 ;$ 6404. 71AB
;3302 ;$ 4119. 0AC2
;3303 ;$ 7084. FFE1
;3304 ;$ 412A. 0BF4
;3305 ;$ 7D05. 8E16
;3306 ;$ 413B. 0D26
;3307 ;$ 8986. 1C4B
;3308 ;$ 414C. 0E58
;3309 ;$ 9606. AA81
;3310 ;$ 415D. 0F8A
;3311 ;$ A287. 38B6
;3312 ;$ 416E. 10BC
;3313 ;$ AF07. C6EC
;3314 ;$ 417F. 11EE
;3315 ;$ BB88. 5521
;3316 ; EXPECTED PRODUCT PATTERN GROUP 2
;3317 ;$ 406E. 00DD
;3318 ;$ AA77. 4411
;3319 ;$ 4086. DE5B
;3320 ;$ 7A32. B470
;3321 ;$ 4096. BC48
;3322 ;$ 1F29. 66D7
;3323 ;$ 40A6. 9A34
;3324 ;$ C420. 493F
;3325 ;$ 40B6. 7821
;3326 ;$ 6917. 2BA6
;3327 ;$ 40C6. 560E
;3328 ;$ 0E0E. 0E0E
;3329 ;$ 40D6. 33FA
;3330 ;$ B304. F075
;3331 ;$ 40E6. 11E7
;3332 ;$ 57FB. D2DD
;3333 ;$ 40FD. DECA
;3334 ;$ 4F6E. 2678
;3335 ;$ 410E. CD51
;3336 ;$ CCAD. F5A3
;3337 ;$ 411E. AB3E
;3338 ;$ 71A4. D80B
;3339 ;$ 412E. 892B
;3340 ;$ 169B. BA72
;3341 ;$ 413E. 6717
;3342 ;$ BB92. 9CDA
;3343 ;$ 414E. 4504
;3344 ;$ 6089. 7F41
;3345 ;$ 415E. 22F1
```

```
;3346 ;$ 0580. 61A9
;3347 ;$ 416E. 00DD
;3348 ;$ AA77. 4410
;3349 ; EXPECTED PRODUCT PATTERN GROUP 3
;3350 ;$ 405D. FFCC
;3351 ;$ 9966. 3300
;3352 ;$ 407B. 995F
;3353 ;$ 5884. E477
;3354 ;$ 408C. 9979
;3355 ;$ 0BD1. CAF7
;3356 ;$ 409B. 6642
;3357 ;$ 6B61. 23B3
;3358 ;$ 40AA. 330B
;3359 ;$ CAF0. 7C6F
;3360 ;$ 40B8. FFD5
;3361 ;$ 2A7F. D52A
;3362 ;$ 40C7. CC9E
;3363 ;$ 8A0F. 2DE6
;3364 ;$ 40D6. 9967
;3365 ;$ E99E. 86A2
;3366 ;$ 40EC. CC95
;3367 ;$ F8F5. 8BBB
;3368 ;$ 4105. 3314
;3369 ;$ 5C0A. 1E99
;3370 ;$ 4113. FFDD
;3371 ;$ BB99. 7755
;3372 ;$ 4122. CCA7
;3373 ;$ 1B28. D010
;3374 ;$ 4131. 9970
;3375 ;$ 7AB8. 28CC
;3376 ;$ 4140. 6639
;3377 ;$ DA47. 8188
;3378 ;$ 414F. 3303
;3379 ;$ 39D6. DA43
;3380 ;$ 415D. FFCC
;3381 ;$ 9966. 32FF
;3382
;3383
```

```
            ;3384  .PAGE "TEST 2A6 RESERVED"
            ;3385 =0
U 10B6, 0018,0039,0980,09F8,0000,10BC  ;3386 TSTXX1: NEWTST[.2]
U 10B7, 0000,003C,0180,0800,0000,128C  ;3387  NOP
            ;3388
```

```
ZZ-ESKAR-V22  DUMMY NEWTST                                                    SEQ 2860
ESKAR66.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40          Page   8

            ;3389  .PAGE "DUMMY NEWTST"
U 128C. 0000,003D,0180,0800,0000,10BC  ;3390 TSRXX3: NEWTST
            ;3391
```

```
ZZ-ESKAR-V22  Line Number Index                                                SEQ 2861
ESKAR66.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40          Page   8
:                             Location / Line Number Index

:Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 0000 - 0FFF Unused
U 1000  1907: 1874 1881= 1897= 1898= 1899= 2183= 2184=
U 1008  1918= 1919= 1875 2000= 2011= 2012= 1876 2001=
U 1010  2042= 2043= 2049= 2050= 2080= 2081= 1877 2460=
U 1018  2101= 2102= 2104= 2105= 2122= 2123= 1878 2461=
U 1020  2124= 2125= 2146= 2147= 2148= 2149= 1879 2482=
U 1028  2165= 2166= 2167= 2168= 2181= 2182= 1880 2483=
U 1030  2203= 2204= 2213= 2214= 2264= 2265= 2267= 2268=
U 1038  2269= 2270= 2272= 2273= 2286= 2287= 2791= 2792=
U 1040  2412= 2413= 2422= 2423= 2428= 2429= 2447= 2448=
U 1048  2457= 2458= 2471= 2472= 2473= 2474= 1939: 1900
U 1050  2479= 2480= 2489= 2490= 2493= 2494= 2496= 2497=
U 1058  2584= 2585= 1942: 1901 2594= 2595= 1945: 1902
U 1060  2600= 2601= 2620= 2621= 2635= 2636= 2637= 2638=
U 1068  2642= 2643= 2646= 2647= 2649= 2650= 2998= 2999=
U 1070  2734= 2735= 2746= 2747= 2752= 2753= 2774= 2775=
U 1078  2798= 2799= 2812= 2813= 2824= 2828= 2829= 2830=
U 1080  2840= 2841= 2844= 2845= 2848= 2849= 2852= 2853=
U 1088  2855= 2856= 2942= 2943= 2953= 2954= 2959= 2960=
U 1090  2981= 2982= 3005= 3006= 3019= 3020= 3032= 3034=
U 1098  3035= 3036= 3046= 3047= 3050= 3051= 3054= 3055=
U 10A0  3058= 3059= 3061= 3062= 3145= 3146= 3164= 3168=
U 10A8  3179= 3180= 3192= 3194= 3195= 3197= 3215= 3216=
U 10B0  3222= 3223= 3226= 3227= 3238= 3239= 3386= 3387=
U 10B8  1903 1904 1905 1906 1916 1917 1920 1921
U 10C0  1922 1923 1924 1925 1926 1927 1928 1929
U 10C8  1930 1931 1932 1933 1934 1935 1936 1937
U 10D0  1938 1940 1941 1973 1974 1980 1981 1982
U 10D8  1996 1997 1998 1999 2009 2010 2013 2020
U 10E0  2021 2023 2024 2027 2028 2030 2031 2032
U 10E8  2034 2035 2037 2038 2039 2045 2046 2052
U 10F0  2053 2054 2063 2064 2065 2066 2067 2068
U 10F8  2069 2082 2083 2084 2085 2086 2087 2088
U 1100  1861: 1862: 1863: 1864: 1865: 1866: 1867: 1868:
U 1108  1869: 2089 2090 2091 1870: 1871: 1872: 1873:
U 1110  2092 2093 2103 2106 2107 2108 2127 2129
U 1118  2130 2131 2132 2150 2151 2152 2153 2154
U 1120  2155 2156 2170 2171 2188 2189 2190 2191
U 1128  2193 2194 2206 2208 2210 2215 2216 2217
U 1130  2218 2219 2220 2221 2222 2228 2229 2238
U 1138  2239 2240 2241 2242 2243 2244 2245 2246
U 1140  2247 2248 2249 2250 2251 2262 2263 2266
U 1148  2271 2274 2288 2289 2290 2296 2297 2298
U 1150  2299 2300 2301 2302 2303 1965: 2304 2305
U 1158  2317 2318 2319 2320 2321 2322 2323 2324
U 1160  2325 2326 2327 2328 2329 2330 2414 2415
U 1168  2416 2417 2418 2419 2420 2421 2424 2427
U 1170  2430 2431 2432 2436 2437 2438 2439 2440
U 1178  2441 2442 2443 2449 2450 2451 2452 2453
U 1180  2454 2455 2456 2459 2462 2463 2464 2465
U 1188  2466 2467 2475 2476 2477 2478 2481 2484
U 1190  2485 2486 2487 2488 2491 2492 2495 2586
U 1198  2587 2588 2589 2590 2591 2592 2593 2596
```

```
;Location 0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

U 11A0  2599 2602 2603 2604 2605 2609 2610 2611
U 11A8  2612 2613 2614 2615 2616 2622 2623 2624
U 11B0  2625 2626 2627 2628 2629 2630 2631 2639
U 11B8  2640 2641 2644 2645 2648 2736 2737 2738
U 11C0  2739 2740 2741 2742 2743 2744 2745 2748
U 11C8  2751 2754 2755 2756 2757 2761 2762 2763
U 11D0  2764 2765 2766 2767 2768 2769 2770 2776
U 11D8  2777 2778 2779 2780 2781 2785 2786 2787
U 11E0  2788 2789 2790 2793 2800 2801 2802 2803
U 11E8  2804 2808 2809 2810 2811 2814 2815 2816
U 11F0  2817 2818 2819 2820 2831 2832 2833 2834
U 11F8  2835 2836 2837 2838 2839 2842 2843 2846
U 1200  2847 2850 2851 2854 2944 2945 2946 2947
U 1208  2948 2949 2950 2951 2952 2955 2958 2961
U 1210  2962 2963 2964 2968 2969 2970 2971 2972
U 1218  2973 2974 2975 2976 2977 2983 2984 2985
U 1220  2986 2987 2988 2992 2993 2994 2995 2996
U 1228  2997 3000 3007 3008 3009 3010 3011 3015
U 1230  3016 3017 3018 3021 3022 3023 3024 3025
U 1238  3026 3027 3028 3037 3038 3039 3040 3041
U 1240  3042 3043 3044 3045 3048 3049 3052 3053
U 1248  3056 3057 3060 3147 3148 3149 3150 3151
U 1250  3152 3153 3154 3155 3156 3157 3158 3159
U 1258  3160 3161 3162 3163 3169 3170 3171 3172
U 1260  3173 3174 3175 3181 3182 3183 3184 3185
U 1268  3186 3187 3188 3198 3199 3200 3201 3202
U 1270  3203 3207 3208 3209 3210 3211 3212 3213
U 1278  3214 3217 3218 3219 3220 3221 3224 3225
U 1280  3229 3230 3231 3232 3233 3234 3235 3236
U 1288  3237 3240 3241 3242 3390
U 1290 - 12A7 Unused
U 12A8    1966:
U 12B0 - 13EF Unused
U 13F0  1949: 1950: 1951: 1952: 1953: 1954: 1955: 1956:
U 13F8  1957: 1958: 1959: 1960: 1961: 1962: 1963: 1964:
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2863
ESKAR66.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40     Page   8

            Words not
            in bounds
 ESK00DEF           0
 ESK00MAC           0
 ESKARMAC           0
  ESKAR66         670

Used               670
Remaining

Total microwords used in memory U: 670
Total microwords remaining in memory U: 0
Highest address used in memory U: 13FF (hex)
```

```
ZZ-ESKAR-V22  Line Number Index                                          SEQ 2864
ESKAR66.MCR                    MICRO2  1P(00)    26-JUL-85  17:11:40     Page   8


; The files used in this assembly are:
ESK00DEF.MIC
ESK00MAC.MIC
ESKARMAC.MIC
ESKAR66.MIC


Pass 1 warnings:    0 Pass 2 warnings:    0
 Unsplit Binary Lines:    0
Pass 1 errors:      0 Pass 2 errors:      0
```